We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter 6

Multimodal Bayesian Network for Artificial Perception

Diego R. Faria, Cristiano Premebida, Luis J. Manso,
Eduardo P. Ribeiro and Pedro Núñez

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.81111

Abstract

In order to make machines perceive their external environment coherently, multiple sources of sensory information derived from several different modalities can be used (e.g. cameras, LIDAR, stereo, RGB-D, and radars). All these different sources of information can be efficiently merged to form a robust perception of the environment. Some of the mechanisms that underlie this merging of the sensor information are highlighted in this chapter, showing that depending on the type of information, different combination and integration strategies can be used and that prior knowledge are often required for interpreting the sensory signals efficiently. The notion that perception involves Bayesian inference is an increasingly popular position taken by a considerable number of researchers. Bayesian models have provided insights into many perceptual phenomena, showing that they are a valid approach to deal with real-world uncertainties and for robust classification, including classification in time-dependent problems. This chapter addresses the use of Bayesian networks applied to sensory perception in the following areas: mobile robotics, autonomous driving systems, advanced driver assistance systems, sensor fusion for object detection, and EEG-based mental states classification.

Keywords: Bayesian networks, machine learning, multimodal robotic perception

1. Introduction

Bayesian networks (BNs) allow a tractable graph-based representation for probabilistic reasoning (or inference), under uncertainty, about a given problem or domain. A recurrent problem in robotics is to reason about the class of an object in the environment, given evidence (from sensors, e.g. RGB-D cameras), and probabilistic models (e.g. probability outputs of a

IntechOpen

© 2018 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. CC BY

classifier) in the domain that represents the problem. For example, a robot would be needed to detect and then recognise a particular type of object (such as a mug) in a given place (e.g. kitchen) [1]. Another example would be an autonomous vehicle that has to detect road users; hence, the object categories of interest would be pedestrian, cyclist, car and van, bus and truck, and motorised two-wheelers.

The topology, or structure, of a BN graph is the first step in solving the problem, and it should provide the relationship (dependencies represented by links) among the nodes (variables in the problem domain). The next step is to define the conditional probabilities for the nodes and, then, the joint probability of the BN has to be considered in order to allow computing the posterior probability of the form P(Class | Evidences), i.e. *a posteriori* of the class, or category, given the evidences from a set of sensor-based models [1].

In this chapter, we will address BN with similar topologies to the one illustrated in **Figure 1**. The structure shown in **Figure 1** is a 'common effect' chain [2], which means that all parent nodes contribute to the node *C* designated by the 'class'. The node *C* is the label variable and takes values such as: $C = \{person, non\text{-}person\}$, or $C = \{1, 0\}$, or in multiple class case, $C = \{mug, spoon, knife, fork, plate, can\}$ or $C = \{concentrated, relaxed, neutral\}$. The evidence nodes, as illustrated in **Figure 1**, provide probability values per class of interest; thus, such nodes are modelled by a classifier (e.g. convolutional neural network [CNN], SVM, and Bayes classifier). The node called 'context' might represent evidence from the environment, or information shared by the infrastructure (e.g. cameras mounted on the scenario), or any other evidence not directly related to a given learning classifier using data/features from sensors onboard the robot.

The remainder of this chapter is organised as follows: Section 2 briefly describes the use of BNs for supervised classification problems. Use cases on object manipulation, pedestrian classification, and EEG-based Mental State Classification are described in Sections 3–5, respectively. Finally, Section 6 presents a summary and remarks.

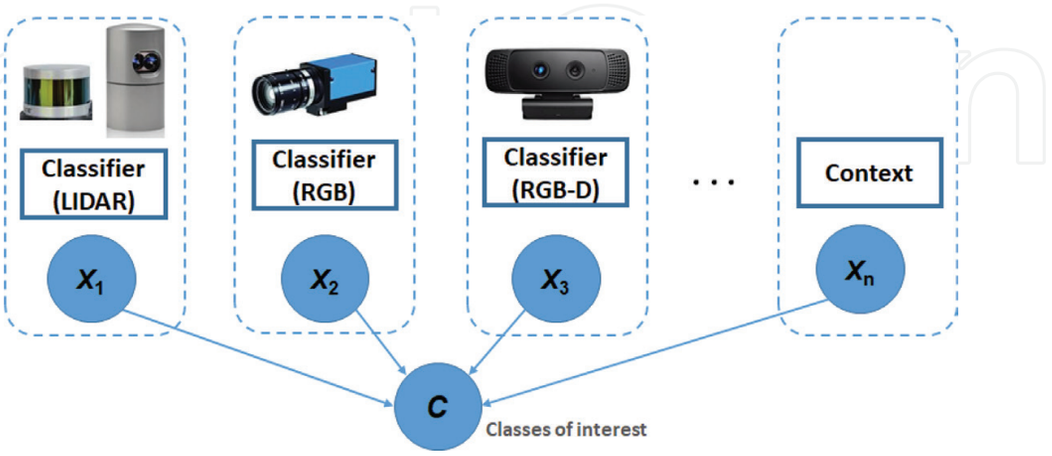


Figure 1. Topology of a BN where all the parent nodes $\{X_1, X_2, \ldots, X_n\}$ contribute to a common effect, which is the set of classes (node *C*) of interest in a given robotic domain.

2. Bayesian networks for supervised classification

In a more general and high-level perspective, a BN is characterised by **nodes** that represent a finite set of random variables, i.e. a variable/function whose outputs outcome from a random measured process (belonging to the domain of interest) and **links** (i.e. directed arcs) that represent the direct dependencies between the nodes. Hereafter, the link dependencies will assume the form of conditional probabilities. By examining **Figure 1**, we can see that each node $X_i, i = 1, \ldots, n$, is conditionally independent of all the nodes, while the node C is conditionally dependent of all its parent nodes $X_1, X_2, \ldots, X_n$. This is a simple BN structure where the nodes represent learning model—in the form of supervised classifiers. On the other hand, which is common in some classification problems, the nodes may represent features as extracted from observed data.

Let $P(X \mid C)$ be the outcome of a learning classifier given the observed/measured sensor data. The variable X can represent a learned model, using a probabilistic classifier, based on supervised and measured data from a camera, a LIDAR, an RGB-D sensor, or a combination of multimodality data. We could represent the conditional probability as $P(X, \theta \mid C)$ to explicitly show that the output also depends on a learning model (here represented by θ).

In a nutshell and considering the use cases described in the sequel, BNs are used to express the joint probability of events (represented by the nodes) that model a classification system where the relationships between events are expressed by conditional probabilities. Given the observations/measurements (evidence) and prior knowledge, statistical inference is accomplished using the Bayes' theorem. The goal is to calculate the posterior $P(C \mid X_i)$ of the set of classes given the evidential nodes $X_i, i = 1, \ldots, n$. So, inference in supervised classification applications aims to estimate the probability of the classes given the class-conditional probabilities, priors and observations.

In this work, in one of the approaches presented, we consider BN structures where the sensory data are transformed to a feature space which is then feed into a trained classifier. The classifier is assumed to output a class-conditional probability which is then used to calculate the *a posteriori*. When multiple sensors are considered, the conditional independency property between sensors will be satisfied, for example: $P(X_{sensor1} \mid X_{sensor2}, X_{sensor3}) = P(X_{sensor1})$. The following sections will present three different study cases where BN are employed as a supervised classifier.

3. Use case in object shape representation through in-hand exploration

Accurate modeling of the world (environment and its components) is important in autonomous robotics applications. More precisely, for grasping applications dealing with objects used in everyday tasks, the object information (intrinsic and extrinsic) acquired before the robot executes a task is crucial for grasp strategies. The object geometry (size and shape) plays an important role in such applications, where its representation is also valuable for classification into a class of known objects and also for identification of regions on the object surface

proper for a stable grasp. Since the robotic end-effector usually relies on the knowledge of object geometry to plan or to estimate grasp candidates, the more accurate the geometry of the object, the higher is the likelihood of success when estimating the candidate's grasp for that object. Many techniques can be used to reconstruct and represent an object using different sensors, such as vision-based systems, laser range finders, etc., where the most common is through visual information.

Mapping techniques such as occupancy grid [3, 4] have been used in robotics to describe the environment of mobile robots. Two-dimensional grids have been used for static indoor mapping as shown in [5]. The idea is to estimate the probability of each cell to be occupied or empty after the sensors' observation. Probabilistic volumetric maps are also useful in robotics by providing means of integrating different occupancy belief maps in order to update a central multimodal map using Bayesian filtering. A grid divides the workspace into equally sized voxels, and the edges are aligned with one of the axes of a reference coordinate frame. The coverage of each voxel given the sequence of batches of measurements is modelled through a probability density function. The probabilistic approach for building volumetric maps of unknown environments can also be based on information theory. Each sensor (e.g. vision, laser, etc.) can adopt an entropy gradient-based exploration strategy to define the occupied regions (most explored) in the map.

Object in-hand exploration is the procedure of exploring the shape of objects using tactile information and fingers motion around the object surface to reconstruct its shape [6]. In order to acquire the probabilistic representation of an object using a volumetric map, it is necessary to have an *a priori* estimation of the area, where the object is placed for mapping. There are two scenarios in which in-hand exploration can be applied: a static object placed at a specific location or an object being explored in-hand in constant motion (dynamic exploration with moving object). The sensors used for this task is a cyberglove that measures fingers flexure (0–255 range), with six electromagnetic motion sensors (Polhemus sensors), where each sensor provides 6D information (*x, y, z, yaw, pitch,* and *roll*), and tactile sensors in each fingertip and palm (Tekscan pressure sensor) that measure the force (0–255 range). **Figure 2** depicts the experimental setup and sensors used for object in-hand exploration.

When the object exploration is in-hand and the object is moving, then it is needed to perform a registration to map the object displacements into a single frame of reference. We can consider that, for every motion of the object, a local map is built, so that all local maps should be integrated into a global map to have the whole representation of the object shape exploration in the same frame of reference. Knowing the object initial position and the object displacements, we can compute the transformations to have all points in the same frame of reference. Given that the sensor attached to the object has six DoF (x, *y, z, yaw, pitch,* and *roll)*, we can compute the rotation and translation of the object. We compute the rotation matrix of the object in a specific point in time using α = yaw (rotation in z axis), β = pitch (rotation in y), and φ = roll (rotation in x).

To map the point cloud in the same frame of reference, for all points, we find the translation of the fingertip sensor to the object sensor and then we apply the rotation to that point, $p' = R_o t$, where p' is the new position of the 3D point that we are mapping to the same frame of reference of the object sensor; R_o is the rotation matrix 3 × 3 of the object sensor; and t the translation of the fingertip sensor to the object sensor.

http://dx.doi.org/10.5772/intechopen.81111

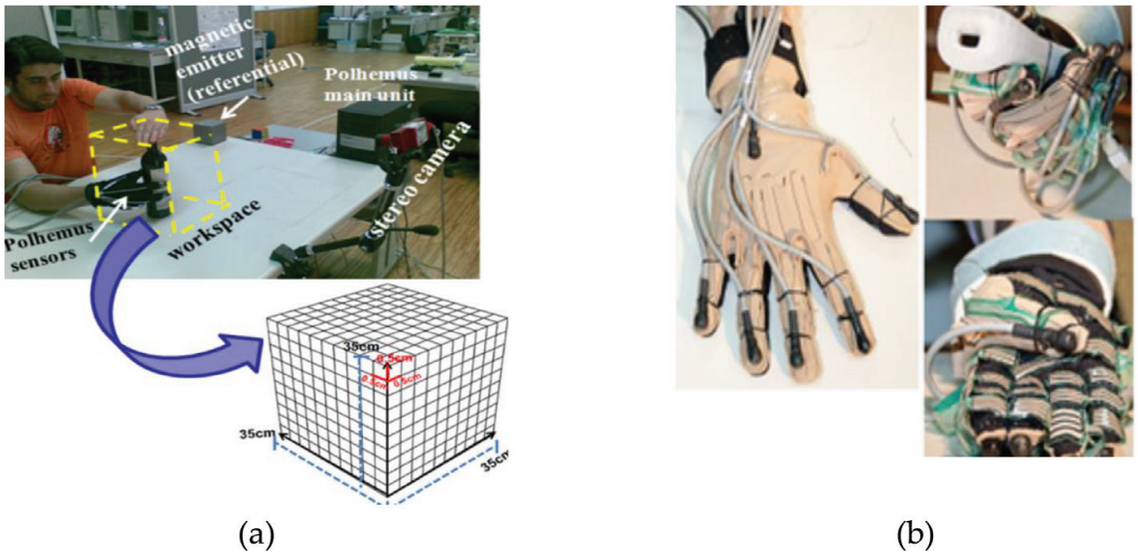


Figure 2. Experimental setup: (a) workspace for mapping (grid 35 cm × 35 cm × 35 cm equally divided, where each voxel is sized with 0.5 cm); (b) Polhemus Liberty Motion Tracking System: magnetic sensors attached to the cyberglove (fingertips and back of the hand).

The Bayesian volumetric map [6] is an occupancy grid, i.e. discrete random fields, wherein each cell has an assigned value, which represents the probability of the cell being occupied. The dimensions of the voxels define the spatial resolution of the representation. The edges of the grid are aligned with one of the axes of the world frame of reference *W*. In this work, the map is a 3D grid comprised of a set of cells $c \in M$, denoted as voxels, wherein each voxel is a cube with edge $\varepsilon \in R$. The voxels divide the workspace into equally sized cubes with volume ε^3. The occupancy of each individual voxel is assumed to be independent from the other voxels occupancy, and thus, *Oc* is a set of independent random variables as follows:

- $c \in M$: Index a cell on the Map;
- $Oc \in [0, 1]$: Probability describing if the cell c is empty or occupied;
- *Zc*: Measurement that influences the cell *c*. It represents the measurements acquired from five sensors, each one returns the 3D location of each finger movement in the map;
- P(*Oc*): Probability distribution of preliminary knowledge describing the occupancy of the cell c, initially as a uniform distribution (0.5 for each state: empty or occupied); and
- P(*Zc*|*Oc*): Probability density function corresponding to the set of measurements that influences the cell c taken from the in-hand exploration measurements. This distribution is computed from the in-hand exploration sensor model.

The knowledge about the occupancy of a voxel *c* in the map *M*, after *Z* measurements received at time *t* from the sensors, is represented by the probability density function $P(Oc|Z_c^t)$. Updating the 3D probabilistic representation of the manipulated object shape upon a new measurement Z^t means updating the probability distribution function $P([Oc = 1]|Z_c^t)$ of the voxel *c* influenced by the measurement *Z* at time *t*. Voxels are influenced by a measurement

Z^t if the location associated with the sample computed from the sensor model P ($Z_c^t | [Oc = 1]$) is contained in that voxel location c. For each voxel c, the set of measurements Z_c^t contains n measurements Z_c influencing a voxel c along the time t. The probability density function of the object shape representation of voxel c given the Z_c measurements influencing such voxel is represented by $P(Z_c^t | [Oc = 1])$. To update the occupancy estimation of a cell in the map, the Bayes rule is applied:

$$P([Oc = 1] | Z_c^t) = \frac{P(Z_c^t | [Oc = 1])P([Oc = 1])}{P(Z_c^t | [Oc = 0])P([Oc = 0]) + P(Z_c^t | [Oc = 1])P([Oc = 1])}, \quad (1)$$

where $P([Oc = 0]) = 1 - P([Oc = 1])$; $P(Z_c^t | [Oc = 1])$ is given by the probability density function computed from the sensor model and $P(Z_c^t | [Oc = 0])$ is a uniform distribution.

Assuming that consecutive measurements Z^t are independent given the cell occupancy, the following expression is obtained:

$$P([Oc = 1] | Z_c^t) = \beta \times P(Oc) \prod_{t=1}^{T} P(Z_c^t | [Oc = 1]), \quad (2)$$

where β is a constant representing a normalization, factor ensuring that the left side of the equation sums up to one over all Oc.

The cells occupancy in the map are probabilities that are updated over time as long as the sensors measurements are active. At the end of the in-hand exploration of the object, the cells are allowed to represent only two states: occupied or empty, $Oc \in [0, 1]$, so that a threshold is used for each cell to consider one of the two states:

$$Oc = \{0, P(Z_c^t) < 0.5\ 1, P(Oc | Z_c^t) \geq 0.5\}. \quad (3)$$

Figure 3 shows an example of the probabilistic volumetric map and its utility. The map can be used to represent the full model of the object as well as partial volume of the object and contact.

Each magnetic sensor attached to the fingertips returns the 3D coordinates of the finger location based on the sensor frame of reference (source/emitter of the Polhemus Liberty tracking system). The frame rate of each sensor was defined to be up to 15 Hz. During data acquisition, a workspace (35 cm^3) is defined in the experimental area for mapping. The grid space is divided into equally sized voxels (also denoted as cells) of 0.5 cm^3. Due to the size of each cell, relative to the standard deviation of the magnetic tracking sensors measurements (up to 3 mm), inside each cell a 3D isotropic Gaussian probability distribution is defined, $P(Z_c^t | Oc)$, centred at the cell central point with the standard deviation 0.3 cm and mean value equal to the central point coordinates of the cell. In other words, this means that the model attempts to ensure that, upon receiving a measurement from the sensor attached to the fingertip, the closer the finger position is to the centre of a specific cell of the map, the more probable that cell is occupied. Furthermore, during the object surface exploration, the more often that the finger passes through that cell, the cell probability is updated with higher certainty in which

http://dx.doi.org/10.5772/intechopen.81111

that given point position actually belongs to the object surface. The probability that a measurement belongs to a cell is given by a normal distribution using the known sensor position error as the standard deviation and the sensors positions relative to the centre of each cell in the map as follows:

$$P(Oc) = \frac{1}{2\,\Pi^{3/2}\,|\Sigma|^{1/2}} e\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right), \tag{4}$$

where $P(z_c^t | Oc)$ represents the probability distribution of the sensor measurement given a specific cell Oc; $|\Sigma|$ represents the determinant of Σ (sensor noise variation). It can also represent a scalar value. After normalization, it takes the form:

$$P([Oc = 1]) = exp\left(-\frac{(x-u_x)^2 + (x-u_y)^2 + (x-u_z)^2}{2\,\sigma^2}\right), \tag{5}$$

where (x, y, z) are the coordinates of the 3D point on the object surface, and u is the central coordinate of the cell (for each axis). The in-hand exploration of objects can be performed by using the thumb and other fingers, i.e. the occupancy grid can be influenced by them over time, thus, expanding on the model for cell update, the contribution of the sensor on each finger through time can be made explicit on the decomposition as follows:

$$P\left(Z_{thumb}^{t=0}, \ldots, Z_{thumb,}^{T} Z_i^{t=0}, \ldots, Z_i^T, Oc\right) = P(Oc) \prod_{t=0}^{T} P\left(z_{thumb}^t \mid Oc\right) \prod_{i=1}^{N} P\left(z_i \mid Oc\right), \tag{6}$$

where T represents the current time instant and $N = 4$, the remaining four fingers of the hand. This process for updating the cell over time recursively (i.e. initially using the cell probability as a uniform distribution: empty or occupied, and later the cell probability—updated with the Bayes rule—is used as prior for the next update), represents a Bayesian network.

The BN representation of the formalism applied to the decomposition of the joint distribution in which the sensor model was used is shown in **Figure 4**. The plate notation relies on assumptions of duplicated subgraph as many times as the associated repetition number (in

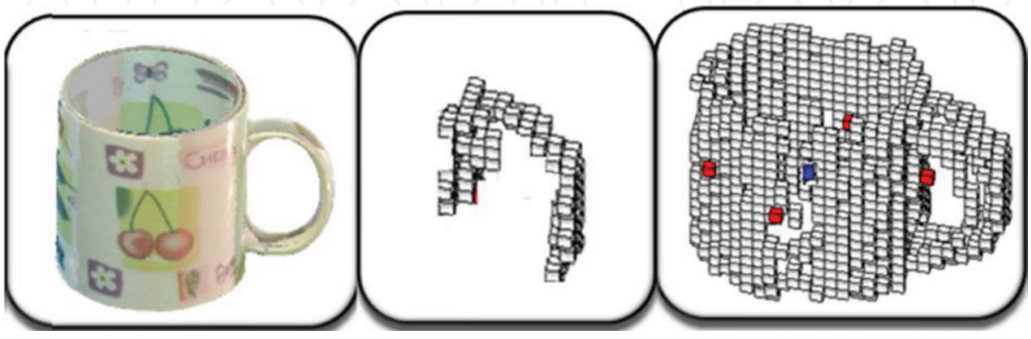


Figure 3. Examples of the Bayesian volumetric map. Left image: real object; middle image: partial volume of the object obtained during in-hand exploration; right image: map of the full object model and contact points overlaid on the object surface (red voxels representing the contact points and blue voxel representing the centroid of the object to define its frame of reference).

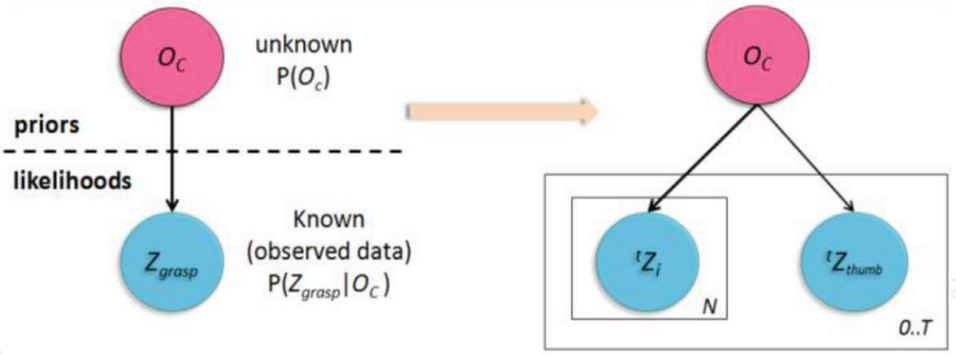


Figure 4. BN for object representation by in-hand exploration using occupancy grid. The left image shows the labels: prior, posterior, and respective distributions, yet not necessary in dynamic BN representations. The variables are defined in terms of their notation and conditional dependence. The instantiation is defined with their parameters and the random variables that support the model are fully described (i.e. their significance and measurable space). The right image shows the plate notation applied to the BN formalism to represent the in-hand exploration of objects, making explicit the contribution of the sensors over time.

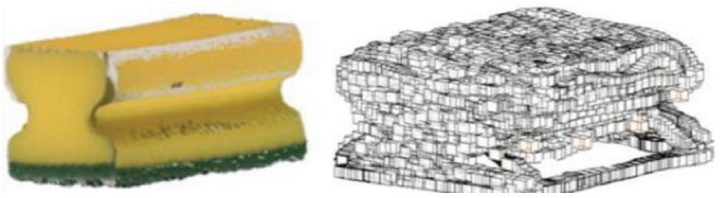


Figure 5. Object representation using the probabilistic volumetric map: sponge and its computed map.

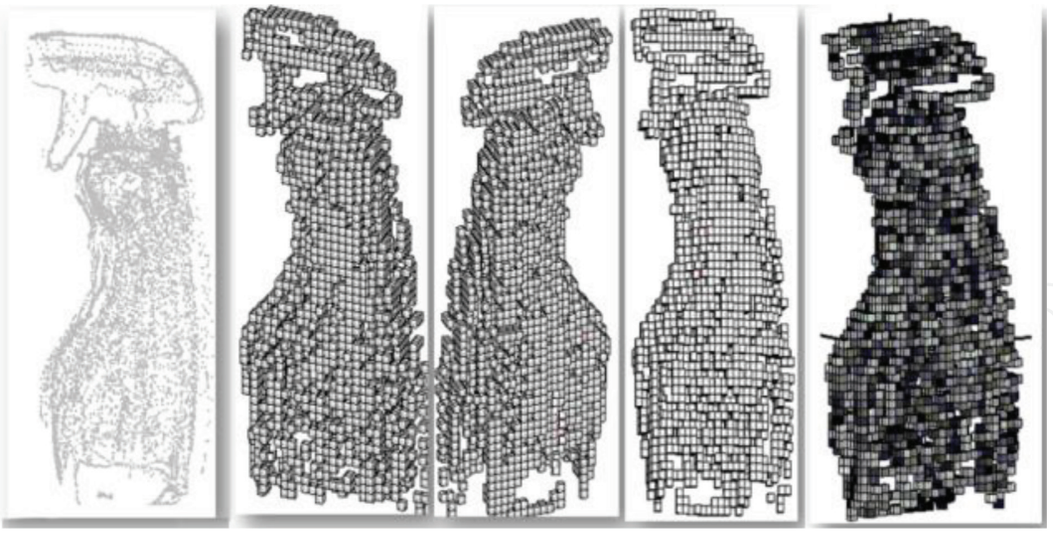


Figure 6. Object shape representation by in-hand exploration of a spray bottle. The first image (left to right) is the raw data (point cloud), next three images are different views of the voxels representation of the object shape, and the last image is the occupancy representation of the cells, the darkest ones represent the lower probabilities (less explored regions).

this particular case the hand fingers); the variables in the subgraph are indexed according to the repetition number; the links that cross a plate boundary are replicated for each subgraph repetition; the distributions are in the joint distribution as an indexed product of the sequence

of variables. Bayesian formalisms for probabilistic model construction and some BN examples of occupancy grid model can also be seen in [6, 7].

Figures 5 and **6** shows different household objects explored in-hand for shape retrieval.

4. Use case in pedestrian classification

A pedestrian detection system is one of the key components in Advanced Driver Assistance Systems (ADAS) and also in autonomous driving vehicles. Recently, pedestrian detection has regained particular attention from academia, automotive industry, and society [8]. In this chapter, pedestrian classification is studied based on a multimodal Bayesian network, where the BN's structure has a node representing the binary class (pedestrian and nonpedestrian) and the parent nodes are represented by machine learning models in the form of supervised classifiers. In terms of sensory data, we will consider a LIDAR sensor as an intermodality technology, which provides range (distance) and reflectance (intensity return). In order to study multimodality between two sensor technologies, a colour (RGB) camera is also considered in the BN. The classifiers are modelled by a deep convolutional neural network (CNN). Data from a LIDAR enter into the CNN classifier in the form of high-resolution distance/depth (DM) and reflectance maps (RMs). Distance and intensity (reflectance) raw data from the LIDAR are transformed to high-resolution (dense) maps as described in [9, 10].

A multimodal BN is then used to combine the likelihoods from CNN-classifiers learned using data from a LIDAR (based on DM and RM) and from a camera. Pedestrian recognition is evaluated on a 'binary classification' dataset created from the KITTI Vision Benchmark Suite, which provides data from a colour camera and from a Velodyne HDL-64E LIDAR. The performance results using the BN are compared with the CNNs having a single modality as input, and against nonlearning rules, namely: minimum, maximum, and average.

We will formulate the classification problem in such a way that the class node (C) of the BN is inferred from the classification nodes (X_{RGB}, X_{DM}, X_{RM}); therefore, the 'full' joint distribution is expressed by:

$$P(C, X_{RGB}, X_{DM}, X_{RM}) = P(C)P(X_{RGB} \mid C)P(X_{DM} \mid X_{RGB}, C)P(X_{RM} \mid X_{DM}, X_{RGB}, C), \tag{7}$$

assuming each classifier node contributes independently to explain C and also assuming the classifiers are independent of each other but not independent of the class so, e.g. $P(X_{DM} \mid X_{RGB}, C) = P(X_{DM} \mid C)$, we can express the class-conditional a posteriori as:

$$P(C \mid X_{RGB}, X_{DM}, X_{RM}) \sim P(C)P(X_{RGB} \mid C)P(X_{DM}, C)P(X_{RM}, C). \tag{8}$$

We will consider the class a-priori probability to be uniform and equally distributed; thus, the probability of being pedestrian or nonpedestrian ($P(C)$) can be dropped out from the equation above. Therefore, the inference problem resumes to a product of the outputs probabilities from the CNN models.

To evaluate the multimodal BN described here, a pedestrian classification dataset was created based on the 2D object-detection dataset of KITTI. The labelled classes are given in the form of 2D bounding box tracklets: 'Pedestrian', 'Car', 'Truck', 'Tram', 'Van', 'Person (sitting)', 'Cyclist' and 'Misc'. The classes were separated in two categories of interest: pedestrian and nonpedestrian, i.e. a binary problem. The number of positives examples is 4487 cropped images (labelled bounding boxes of type 'Pedestrian'), while the negative class has 47,378 cropped images (types: 'Cyclist'. 'Car', 'Person (sitting)' and so on). It was considered 70% for the training set (10% of that for validation) and the remaining 30% for the testing set. **Table 1** gives a summary of the dataset used in this use case.

Among several convolutional neural networks, we opted to use AlexNet CNN architecture with batch normalization in the first two layers and the last layer, the *softmax* activation function with two classes and dropout of 50%. The network was trained from scratch for the pedestrian and nonpedestrian classes [10]. Through the bounding boxes provided by the KITTI dataset, we cropped the objects contained in the depth and reflectance maps images. All objects were resized to the size of 227 × 227 because this is the network input size. The network was trained with the following parameter settings: 30 epochs, batch size equal 64, stochastic gradient descent optimizer with *lr* = 0:001 (learning rate), *decay* = 10 − 6 (learning rate decay over each update), *momentum* = 0.9, and categorical cross-entropy as loss function.

Denoting $P(X_i \mid C)$ the confidence (i.e. the class-conditional probability) yielded by deep models CNN*i* (*i = 1, …, n*), where n is the number of models, CNN1 and CNN2 denote CNN models learned from DM and RM (reflectance), respectively, while CNN3 denotes a model using RGB data. Three nonlearning fusion rules are considered: average (AVE), maximum (MAX), and minimum (MIN). The average rule calculates the simple mean of the CNN-classifiers outputs $F\text{-}ave = \frac{1}{n}\sum_{i=1}^{n} P(X_i \mid C)$. The maximum rule outputs the maximum value over the classifier responses, $F\text{-}max = \max\{P(X_i \mid C)\}$, while the minimum rule is $F\text{-}min = min\{P(X_i \mid C)\}$.

The pedestrian classification results are reported using Precision (Pre), Recall (Rec), and F-score (F1) performance measures, allowing a more detailed and accurate analysis of the results. The F-scores values were obtained considering a threshold of 0.5. A number of pedestrian and nonpedestrian examples are unbalanced, as shown in **Table 1**; thus, F-score is here considered because it is a suitable performance measure for unbalanced cases. The results obtained using the BN and the rules AVE, MAX, and MIN are shown in **Figure 7**.

Summary of dataset for pedestrian classification	
Training set	n# positives = 2827
	n# negatives = 29,849
Validation set	n# positives = 314
	n# negatives = 3316
Testing set	n# positives = 1346
	n# negatives = 14,213

Table 1. Pedestrian dataset.

http://dx.doi.org/10.5772/intechopen.81111

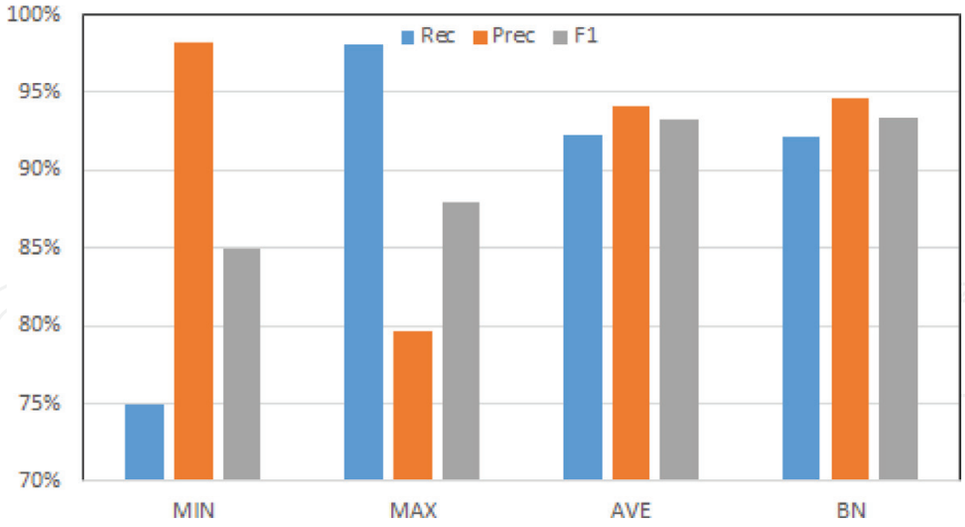


Figure 7. Classification performance, in terms of Pre, Rec, and F-1, considering a multimodal BN in comparison with deterministic rules (Min, Max, Average).

Results show that decision rules like minimum and maximum tend to have poor results, in terms of F-score, compared to the average rule and the multimodal BN. However, the values of Precision and Recall (or True Positive rate) are very high for Min and Max, respectively. The Average and the BN achieved close classification performance in all measures, although the BN's results were slightly better.

5. Use case in EEG-based mental states classification

AI-enabled wearable technology has the ability to enhance the capabilities of today's user-centred devices and analytics toward promoting humans' quality of life and enabling an improved health care by monitoring humans' complex bio-signals, reducing risks, detecting anomalous situations, thus, optimising standards of care. A good example is the EEG-based brain-controlled devices that can serve as powerful aids for severely disabled people in their daily life, especially to help them to move voluntarily. The EEG-based brain-machine interfaces are one of the many alternatives that can be used to interact with devices using the superficial brain activity signals. These signals, called electroencephalograms or EEG for short, convey information regarding the voltage measured by electrodes (dry or wet) placed around the scalp of an individual. Recently, new applications for restoring function to those with motor impairments using EEG-based brain machine interfaces for conveying messages and commands to devices such as robot arm, wheelchair, and any other devices using bio-signals have been developed. A good example where EEG is employed is to detect mental states. The ability to autonomously detect mental states, whether cognitive or affective, is useful for multiple purposes in many domains such as robotics, health care, education, neuroscience, etc. The importance of efficient human-machine interaction mechanisms increases with the number of real life scenarios where smart devices, including autonomous robots, can be applied. One of the many alternatives that can be used to interact with machines is through

superficial brain activity signals. A major challenge in brain-machine interface applications is inferring how momentary mental states are mapped into a particular pattern of brain activity. One of the main issues of classifying EEG signals is the amount of data needed to properly describe the different states, since the signals are complex. The signals are considered stationary only within short intervals, that is, why the best practice is to apply short-time windowing technique in order to detect local discriminative features to meet this requirement.

This section presents how Bayesian inference can be used to classify mental states. The framework consists of (i) statistical and temporal features extraction using time window technique, (ii) attributes selection to keep only the relevant information from the signals, and (iii) Bayesian classification technique to categorise multiple mental states (e.g. relaxed, neutral, and highly concentrated).

5.1. Data acquisition

The sensor Muse Headband was used for data collection. The Muse is a commercial EEG sensing device with five dry-application sensors, one used as a reference point (NZ, at the centre of the forehead) and four (at points TP9, AF7, AF8, TP10, i.e. around the forehead **Figure 8**) to record brain wave activity. To prevent the interference of electromyographic signals, nonverbal tasks that required little to no movement were set. Blinking, though providing interference to the AF7 and AF8 sensors, was neither encouraged nor discouraged to retain a natural state. This was due to the dynamicity of blink rate being linked to tasks requiring differing levels of concentration, and as such, the classification algorithms would take these patterns of signal spikes into account. In addition, subjects were asked not to close their eyes during any of the tasks. Three stimuli were devised to cover the three mental states available from the Muse Headband—relaxed, neutral, and concentrating. A dataset was created after five participants performing the three mental states, where each session lasted 1 minute. The relaxed task had the subjects listening to low-tempo music and sound effects designed to aid in meditation while being instructed on relaxing their muscles and resting. For a neutral mental, a similar test was carried out, but with no stimulus at all, this test was carried out prior to any others to prevent lasting effects of a relaxed or concentrative mental state. Finally, for concentration, the subjects were instructed to follow the 'shell game' in which a ball was hidden under one of the three cups, which were then switched, the task was to try and follow which cup hid the ball. After a short amount of time into the stimulus starting, as to not gather data with an inaccurate class, the EEG data from the Muse Headband were automatically recorded for 60 seconds. The data were observed to be streaming at a variable frequency within the range of 150–270 Hz.

5.2. Feature extraction

Feature extraction and classification of EEG signals are primary goals in brain–computer interface (BCI) applications. One challenging problem when it comes to EEG feature extraction is the complexity of the signal. Nonstationary signals can be observed during the change in alertness and wakefulness, during eye blinking, and also during transitions of mental states. Discriminative features rely on statistical techniques such as mean, standard deviation, autocorrelation, statistical moments of third and fourth order (skewness and kurtosis

http://dx.doi.org/10.5772/intechopen.81111

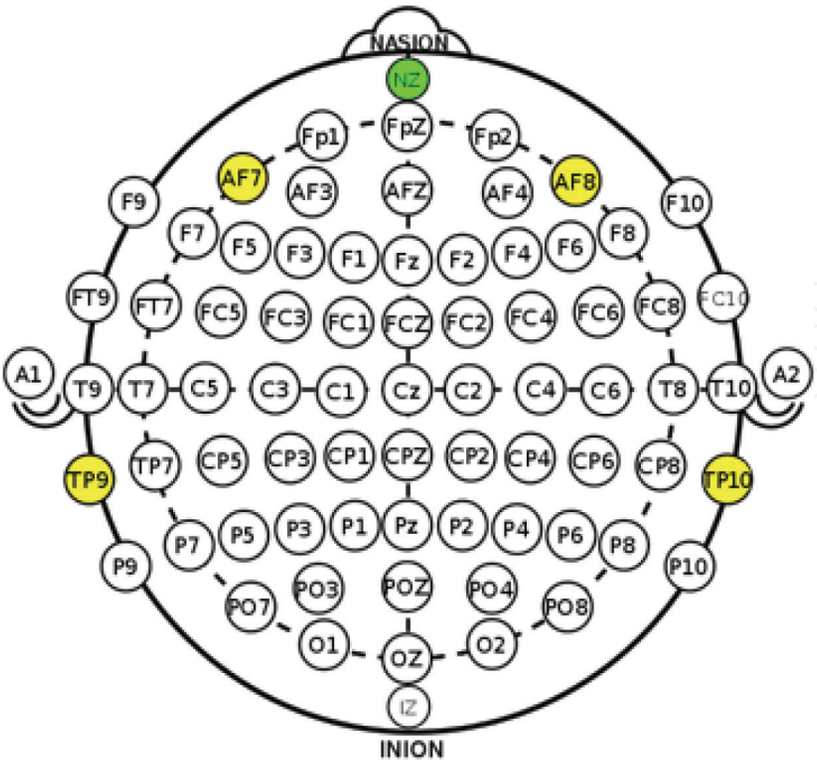


Figure 8. The International 10-20 EEG electrode placement standard [11]. The sensors of the Muse Headband are denoted in yellow. The NZ placement (green) is used as a reference for calibration.

to measure the asymmetry of the data and also the peakedness of the probability distribution of the data), time-frequency based on fast Fourier transform (FFT), Shannon entropy, max-min features in temporal sequences, log-covariance given a set of statistical data, and derivatives of the features from different time instants. These features are computed in terms of the temporal distribution of the signal in a time window of 1 second, with overlap of half second between the sliding windows. Details about the modeling and implementation of the features can be found in [12]. Another important point to compute the features is the signals from the EEG Muse headband. Since it returns five types of signal frequencies (alpha, beta, theta, delta, and gamma), then, we compute all set of features for each signal. The aforementioned set of features for all signals are around 2100 feature values. In order to reduce and optimise the classification performance, feature selection is needed.

5.3. Feature selection

There are various well-known algorithms for features selection in the state of the art. These types of algorithms aim at reducing the number of attributes present in a dataset while retaining a model's predictive accuracy. The following algorithms were used to compare the accuracy performance when used with a Naïve Bayes classifier (NB) and a Bayesian Network (BN): (i) *OneR* calculates error rate of each prediction based on one rule and selects the lowest risk classification [13]; (ii) *Information Gain* assigns a worth to each individual attribute by measuring the information gain with respect to the class (difference of entropy) [14]; and (iii) *Evolutionary*

Algorithm creates a population of attribute subsets and ranks their effectiveness with a fitness function to measure their predictive ability of the class [15]. At each generation, solutions are bred to create offspring, and weakest solutions are killed off in a tournament of fitness.

5.4. Classification

Two models were trained on Bayes' theorem, a formula of conditional probability based on hypothesis H and evidence E. The theorem states that the probability of the hypothesis being true before evidence $P(H)$ is related to the probability of the hypothesis after reading the evidence $P(H \mid E)$ and is given as follows: $P(H \mid E) = \frac{P(E \mid H)P(H)}{\sum_i P(E \mid H)P(H)}$. A simplistic Naive Bayes model has been used, which has a non-cosideration of the relationships between the features models. It uses the maximum a posteriori decision rule $\hat{y} = \operatorname{argmax}_{k \in \{1,\ldots,K\}} P(C^k) \prod_{i=1}^{n} P(x_i \mid C^k)$. A BN (*Bayes Net*) model was also trained. This method generates a probabilistic graphical model via representing probabilities of variables to classes on a directed acyclic graph (DAG) as follows: $P(C^{t-1:t-T} \mid X^{t:t-T}) = \frac{1}{\beta} \prod_{k=t}^{T-t} P(X^k \mid C^k) P(C^k)$. The goal is to infer the current time value of C^t given the data $X^{t:t-T} = \{X^t, X^{t-1}, \ldots, X^{t-T}\}$ and the prior knowledge of the class, which is attained by the a-posteriori probability $P(C^t \mid C^{t-1:t-T}, X^{t:t-T})$. The superscript notation denotes the set of values over a time interval.

5.5. Experimental results

The five generated sets from the original dataset classified by NB and BN are shown in **Table 1**. The most effective model for this EEG dataset using Bayesian inference was the BN along with the *OneR Attribute Selector*, which had a high accuracy of 73.67% using around 2% of the total of features extracted when classifying the data into one of the three mental states. For each test, 10-fold cross-validation was used to train the model. The lowest performance is 54.2% (*Information Gain* dataset with a NB classifier). It is reasonable to assume that the naivety in not considering attribute relationships has led to poorer results. These preliminary results show that a BN can be considered for EEG data classification. However, other methods of classification can achieve better performance with the same set of features. In order to improve the performance, we can adopt the strategy of fusion of multiple classifiers using the Bayes' theorem for fusion as shown in [1, 16] **Table 2** presents the result of Bayesian inference combined with feature selection algorithms. Better results are attained when using OneR algorithm for features selection followed by classification via Bayesian networks.

Dataset	**Accuracy %**		
	Naive Bayes	**Bayesian network**	**Number of selected features (%)**
OneR	**56.30**	**73.67**	44 (2.05)
Information gain	54.20	71.64	31 (1.44)
Evolutionary algorithm	55.04	70.31	99 (4.61)

Table 2. Accuracy of trained models.

6. Summary

Approaches based on Bayesian network (BN) have been described considering three case studies: Bayesian volumetric map for object perception, pedestrian classification for autonomous-vehicles perception and for EEG-based mental states classification. BNs were formulated and applied in supervised pattern classification problems. In all cases, the BNs assumed conditional independence between sensors' modalities or feature models.

In summary, this chapter has addressed BN with examples, where other machine learning techniques were employed and combined with BN to sensory perception in applications related to robotics (multimodal sensor fusion for object detection), advanced driver assistance systems for autonomous driving systems, and EEG-based mental states classification, which can be used to control devices (e.g. robots) or in health-related areas for mental health monitoring.

Acknowledgements

This work has been partially supported by the MICINN Project TIN2015-65686-C5-5-R, by the Extremaduran Government project GR15120, by the FEDER project 0043-EUROAGE-4-E (Interreg V-A Portugal-Spain - POCTEP), and by Fundação Araucária (CONFAP Brazil) with a mobility grant to Dr Diego R. Faria and Professor Eduardo P. Ribeiro to coordinate the project "Stepping-stones to transhumanism: merging EEG-EMG data to control a low-cost prosthetic hand".

Author details

Diego R. Faria[1]*, Cristiano Premebida[2], Luis J. Manso[1], Eduardo P. Ribeiro[3] and Pedro Núñez[4]

*Address all correspondence to: d.faria@aston.ac.uk

1 School of Engineering and Applied Science, Aston University, UK

2 Institute of Systems and Robotics, University of Coimbra, Portugal

3 Department of Electrical Engineering, Federal University of Parana, Brazil

4 School of Technology, University of Extremadura, Cáceres, Spain

References

[1] Premebida C, Faria DR, Nunes UJ. Dynamic Bayesian network for semantic place classification in mobile robotics. In: Autonomous Robots (AURO). Springer; 2017;**41**(5):1161-1172

[2] Korb KB, Nicholson AE. Bayesian Artificial Intelligence. 2nd ed. Boca Raton, FL, USA: CRC Press, Inc.; 2010

[3] Moravec HP. Sensor fusion in certainty grids for mobile robots. AI Magazine. 1988; **9**(2):61-74

[4] Elfes A. Using occupancy grids for mobile robot perception and navigation. IEEE Computer. 1989;**22**:46-57

[5] Thrun S, Burgard W, Fox D. Probabilistic robotics. In: Arkin RC, editor. MIT press; 2005. ISBN 9780262332750

[6] Faria DR. Probabilistic learning of human manipulation of objects towards autonomous robotic grasping [PhD thesis]. Portugal: Department of Electrical and Computer Engineering, University of Coimbra; 2014

[7] Faria DR, Martins R, Lobo J, Dias J. Probabilistic representation of 3D object shape by in-hand exploration. IEEE/RSJ International Conference on Intelligent Robots and Systems; 2010

[8] Ross PE. Uber robocar kills pedestrian, despite presence of safety driver. In: IEEE Spectrum. Available from: https://spectrum.ieee.org [Accessed: June 2018]

[9] Premebida C, Garrote L, Asvadi A, Pedro Ribeiro A, Nunes U. High-resolution LIDAR-based depth mapping using bilateral filter. IEEE International Conference on Intelligent Transportation Systems ITSC; 2016

[10] Melotti G, Premebida C, Goncalves N, Nunes U, Faria DR. Multimodal CNN pedestrian classification: A study on combining Lidar and camera data. 21th IEEE International Conference on Intelligent Transportation Systems (ITSC), 2018

[11] Klem GH, Lüders HO, Jasper HH, Elger C. The ten-twenty electrode system of the International Federation. The International Federation of Clinical Neurophysiology. Electroencephalography and Clinical Neurophysiology. Supplement. 1999;**52**:3-6

[12] Bird JJ, Manso LJ, Ribeiro EP, Ekart A, Faria DR. A study on mental state classification using EEG-based brain-machine interface. In: 9th IEEE International Conference on Intelligent Systems; 2018

[13] University of Waikato. OneR [online] Weka.sourceforge.net. Available from: http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/OneR.html [Accessed: August 2018]

[14] University of Waikato. InfoGainAttributeEval [online] Weka.sourceforge.net. Available from: http://weka.sourceforge.net/doc.dev/weka/attributeSelection/InfoGainAttributeEval.html [Accessed August 9, 2018]

[15] Back T. Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Press; 1996

[16] Premebida C, Faria DR, de Souza FA, Nunes UJ. Applying probabilistic mixture models to semantic place classification in mobile robotics. In: IEEE International Conference on Intelligent Robots and Systems; 2015. pp. 4265-4270