Bayesian Programming: life science modeling and robotics applications

Pierre Bessière, Francis Colas

Bayesian Programming: life science modeling and robotics applications

Pierre Bessiere[1] and Francis Colas[2]

[1] CNRS - Grenoble University - FRANCE Pierre.Bessiere@imag.fr
[2] College de France - Paris - FRANCE Francis.Colas@college-de-france.fr

How to use an incomplete and uncertain model of the environment to perceive, infer, decide and act efficiently? This is the challenge both living and artificial cognitive systems have to face.

Logic is by nature unable to deal with this question. The subjectivist approach to probability is an alternative to logic specifically designed to face this challenge.

In this paper we introduce Bayesian Programming, a methodology, a formalism and an inference engine to build and compute probabilistic models. The principles are illustrated with two examples: *modeling human perception of structure from motion* and *playing to train a video game avatar*.

1 Probability as an alternative to logic for rational sensory–motor reasoning and decision

The *subjectivist* approach to probability (often partially improperly called "Bayesian approach") proposes probability theory as an alternative to logic for rational reasoning in presence of incompleteness and uncertainty [Jaynes, 2003].

A radical proposition indeed, as logic is the founding paradigm of mathematic, of the scientific methodology, of computer science and technology and, possibly, the only common denominator of the different theories of brain and cognition.

The rationale for this proposition is that both living creatures and artificial cognitive systems have to face the same fundamental difficulty: *how to use an incomplete and uncertain model of their environment to perceive, infer, decide and act efficiently?* Indeed, any model of a real phenomenon is *incomplete*. Hidden variables, not taken into account in the model, influence the phenomenon. The effect of these hidden variables is that the model and the phenomenon never behave exactly alike. *Uncertainty* is the direct and unavoidable consequence of incompleteness. No model may foresee exactly the future observations of a phenomenon, as these observations are biased by the hidden variables. No model may either predict exactly the consequences of its decisions.

The subjectivist approach deals with incompleteness and uncertainty with a two steps process: *learning* and *inference* (see Figure 1).

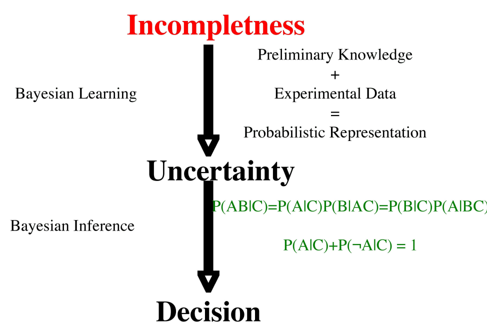

Fig. 1. The subjectivist approach to incompleteness

Learning: Learning transforms the irreducible incompleteness into quantified uncertainty (i.e. probability distributions). These distributions result from both the *preliminary knowledge* of the reasoning subject and the *experimental data* coming from the observation of the phenomenon.

The preliminary knowledge, even imperfect and incomplete, is relevant and provides interesting hints about the observed phenomenon. It is not a fixed and rigid model purporting completeness but a gauge, with free parameters, waiting to be molded by the experimental data.

The experimental data obtained from the physical interaction with the environment reflects all the complexity of this interaction. This includes the effect of the hidden variables that are not taken into account by the preliminary knowledge. Learning sets the values of the free parameters of the preliminary knowledge using the experimental data. Thus, the influence of the hidden variables is taken into account and quantified.

The more accurate and pertinent (i.e. the less incomplete) the preliminary knowledge is, the less uncertain and the more informational the learned distributions are.

Inference: Inference is performed with the probability distributions obtained by the first step. To do so, we only require the two basic rules of Bayesian inference (Bayes theorem and the normalization rule). These two rules are to Bayesian inference what the resolution principle is to logical reasoning [Robinson, 1979], they are sufficient to perform any inference on discrete probability distributions. These inferences may be as complex and subtle as those achieved with logical inference tools. Indeed, it can be shown that logical inference is a particular case of probabilistic inference [Jaynes, 2003].

2 Bayesian Programming formalism and automation

The necessary information to completely specify a probabilistic model is made of the relevant variables, the conditional dependencies between these variables, the parametric forms of the associated probability distributions and the values of these forms' parameters.

Such models are usually presented with different graphic forms (Bayesian Networks by Pearl [1988] or graphical models by Frey [1998]). The relevant variables are nodes, their conditional dependencies are vertices between these nodes, and the parametric forms are associated to each node. As probability is an alternative to logic, it is also possible to use an algebraic formalism to define the probabilistic models. We proposed the *Bayesian Programming* formalism to do so [Lebeltel et al., 2004].

A Bayesian program (BP) is made of two parts:

- a *description* that is the probabilistic model of the studied phenomenon or the programmed behaviour; and
- a *question* that specifies an inference problem to solve using this model.

A description itself contains two parts:

- a *specification* part that formalizes the knowledge of the programmer; and
- an *identification* part in which the free parameters are learned from experimental data.

Finally, the specification is constructed from three parts:

- the selection of *relevant variables* to model the phenomenon;
- a *decomposition*, whereby the joint distribution on the relevant variables is expressed as a product of simpler distributions exploiting conditional independence between variables; and
- the *parametric forms* in which either a given mathematical function or a question to another BP is associated with each of the distributions appearing in the decomposition.

Two examples of a Bayesian Program (see figures 2 and 6) are given in the next sections, where the methodology to build them is also explained.

Furthermore, we developed ProBT®[3] a C++ API which given a BP automate all the necessary computation to get the probability distribution answering the question.

3 Perception of structure from motion

Perception is often stated as the inverse problem of infering properties on the environment starting from sensations given by the senses and some uncertain

[3] This API is available as a commercial product from the ProBAYES company (www.probayes.com) and for free for research and teaching purposes on the Bayesian-Programming.org site (www.bayesian-programming.org).

rules of what should the sensations be for a given state of the environment. The perception of three-dimensional structure from motion (SFM) is one such example that has been studied [Helmholtz, 1867, Wallach and O'Connell, 1953]. However SFM is ill-posed as an infinite number of combinations of geometry and motion can lead to the same optic flow. The first hypothesis introduced to reduce this ambiguity is that the optic flow is produced by a rigid object in motion [Ullman, 1979, Koenderik, 1986]. However recent experiments have shown that the same optic flow could be interpreted differently depending on the motion of the observer. This lead to the proposal of the stationarity hypothesis, which states that the motion of the object is probably small in the allocentric reference frame [Wexler et al., 2001, Wexler, 2003].

3.1 Bayesian model

In this section, we take the example of the perception of a moving planar surface. This general set-up involves a monocular participant placed in front of a monitor in the dark. The stimulus is a set of light points on a dark background depicting a planar patch with a random-dot texture. The participants stare at a fixation cross and the movement of the head of the participants are recorded so as to control the optic flow presented. The task consists in reporting the configuration of the plane by aligning a grid presented on the screen to the mean configuration of the plane perceived.

We define a Bayesian model of an observer submitted to this experiment. It takes as input the optic flow and the self-motion of the observer, and delivers a judgement on the plane configuration. This model is based on the following three hypotheses:

- (H1) rigidity: the optic flow perceived is probably caused by a rigid object in relative motion,
- (H2) stationarity: the movement of this object is probably small,
- (H3) independence between the configuration of the object, the motion of the object and the motion of the observer.

The first two hypotheses are justified in the literature whereas the third one is a common assumption made explicit in our model.

Variables: Within the Bayesian Programming formalism, the first step in the definition of a model is the choice of the relevant variables. This choice is done according to the aim of the model and the hypotheses involved.

In our case, we wish to extract the configuration of a plane from the observation of the optic flow. Therefore we have at least a variable Θ to represent the configuration and a variable $\mathbf{\Phi}$ to represent the optic flow. More precisely, Θ will be the tilt angle of the plane; that is, the angle between the vertical and the intersection between the object plane and the fronto-parallel plane. The optic flow $\mathbf{\Phi}$ will be parametrized by the Taylor series of the velocity field of the projected dots (see Longuet-Higgins [1984]).

The model has to take into account the self-motion of the observer. This is represented with the variable $\mathbf{M}$ that is the 3D-vector of translation of the observer in an allocentric reference frame.

Finally, the rigidity hypothesis H1 involves the relative motion of the plane with respect to the observer and hypothesis H2 involves the absolute motion of the plane with respect to an allocentric reference frame. Thus, we consider respectively the variables $\mathbf{X}_R$ and $\mathbf{X}_A$ for both the relative and absolute motion of the plane.

Decomposition In order to specify the joint distribution, we use the conditional independencies implied by our hypotheses.

Hypothesis (H1) states that the optic flow does not depend on anything else than the object and its relative motion with respect to the observer. This writes:

$$P(\mathbf{\Phi} \mid \Theta\ \mathbf{M}\ \mathbf{X}_R\ \mathbf{X}_A) = P(\mathbf{\Phi} \mid \Theta\ \mathbf{X}_R) \tag{1}$$

On the other hand, hypothesis (H3) states that the configuration of the object, the motion of the object and the motion of the observer are independent. This writes:

$$P(\Theta\ \mathbf{M}\ \mathbf{X}_R\ \mathbf{X}_A) = P(\Theta)P(\mathbf{M})P(X_A)P(\mathbf{X}_R \mid \mathbf{M}\ \mathbf{X}_A) \tag{2}$$

Using Bayes' rule on the joint distribution over all the variables of our model, we write the decomposition:

$$P(\mathbf{\Phi}\ \Theta\ \mathbf{M}\ \mathbf{X}_R\ \mathbf{X}_A) = P(\Theta)P(\mathbf{M})P(\mathbf{X}_A)P(\mathbf{X}_R \mid \mathbf{M}\ \mathbf{X}_A)P(\mathbf{\Phi} \mid \Theta\ \mathbf{X}_R) \tag{3}$$

Parametric forms: The decomposition involves five factors that we have to specify in order to complete the joint distribution over all the variables of the model:

- $P(\Theta)$: as none of our hypotheses involves this prior, we use the maximum entropy distribution which yields a uniform distribution over the angle Θ;
- $P(\mathbf{M})$: this factor is not used in the inference (see expression 5) and can be left unspecified;
- $P(\mathbf{X}_A)$: the stationarity hypothesis (H2) states that the absolute motion is more probably small, thus this distribution is chosen as a Gaussian distribution with the null vectors as mean and a given covariance matrix.
- $P(\mathbf{X}_R \mid \mathbf{M}\ \mathbf{X}_A)$: this factor expresses the reference frame change between absolute and relative motion. More precisely, this is a Dirac distribution on the exact relative motion computed from the given motion of the observer and absolute motion of the object: $\mathbf{x}_R(\mathbf{M}, \mathbf{X}_A)$.
- $P(\mathbf{\Phi} \mid \Theta\ \mathbf{X}_R)$: the rigidity hypothesis (H1) states that the optic flow is probably due to a rigid object. Therefore we specify this factor as a Gaussian distribution, centered on the theoretical optic flow for a plane in this given configuration Θ and for this given relative motion $\mathbf{X}_R$. The covariance matrix states how much we expect the flow to differ from a rigid one.

Question and inference: The task of the observer is evaluate the configuration of the plane given the optic flow and its own motion. In probabilistic terms, this means computing:

$$P(\Theta \mid \boldsymbol{\phi}\ \mathbf{m}) \tag{4}$$

With the joint distribution over all our variables, standard Bayesian inference yields the following expression to answer this question:

$$P(\Theta \mid \boldsymbol{\phi}\ \mathbf{m}) \propto P(\Theta) \sum_{\mathbf{x}_A} P(\mathbf{x}_A) P(\boldsymbol{\phi} \mid \Theta\ \mathbf{x}_R(\mathbf{m}, \mathbf{x}_A)) \tag{5}$$

The model is summarized in figure 2.

Program
Description
Specification

Relevant Variables:
$\mathbf{\Phi}$, Θ, $\mathbf{M}$, $\mathbf{X}_R$, and $\mathbf{X}_A$
Decomposition:
$P(\mathbf{\Phi}\ \Theta\ \mathbf{M}\ \mathbf{X}_R\ \mathbf{X}_A) =$
$P(\Theta)P(\mathbf{M})P(\mathbf{X}_A)P(\mathbf{X}_R \mid \mathbf{M}\ \mathbf{X}_A)P(\mathbf{\Phi} \mid \Theta\ \mathbf{X}_R)$
Parametric Forms:
$P(\Theta)$: uniform;
$P(\mathbf{M})$: unused;
$P(\mathbf{X}_A)$: Gaussian centered on no motion;
$P(\mathbf{X}_R \mid \mathbf{M}\ \mathbf{X}_A)$: Dirac on $\mathbf{x}_R(\mathbf{M}\ \mathbf{X}_A)$;
$P(\mathbf{\Phi} \mid \Theta\ \mathbf{X}_R)$: Gaussian on the optic flow.
Identification:
Choice of the covariances with respect to experimental results.
Question:
$P(\Theta \mid \boldsymbol{\phi}\ \mathbf{m})$

Fig. 2. Bayesian model of shape perception from motion

3.2 Results: stationarity and rigidity

To validate the model, we compared its results with experimental results from six different psychophysical experiments [Colas et al., 2007]. Here, we present the example of a conflict between both the rigidity and stationarity hypothesis.

Description of the experiment: The experimental results are taken from Wexler et al. [2001]. The aim is to investigate the relative importance of both stationarity and rigidity by presenting stimuli with two main interpretations: one more rigid and a second more stationary.

More precisely, the optic flow of a rotating plane can be similar to a the optic flow of a plane in another orientation with both a relative translation and rotation (see figure 3). When the optic flow of the first plane is presented, the second can sometimes be perceived, even if this alternative solution is less rigid (not exactly the same optic flow) and less stationary (one translation and one rotation compared to only the rotation). However, if the observer is in translation in the opposite direction, the relative translation of the object in the alternative solution and the translation of the observer can cancel out the translation of the object in the absolute reference frame. In this case, the alternative is still less rigid than the original solution but is more stationary.

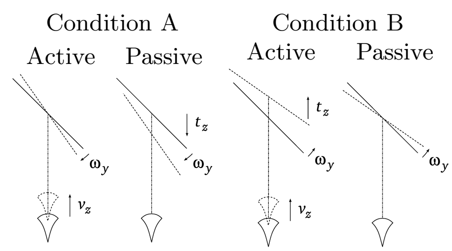

Fig. 3. Summary of experimental conditions. The optic flow is exactly the same for conditions A-active and A-passive, and for conditions B-active and B-passive.

In this experiment, the observer is asked either not to move the head, called immobile condition, or to move the head in a sagittal translation, called active condition. The simulated plane can either be in relative translation with respect to the observer, condition A, or at a fixed distance, condition B. Immobile/active and condition A and B are crossed to produce the four experimental conditions sumarized in figure 3. The velocity of the plane is set according to the motion of the observer so that the orientations of both simulated and alternative planes are orthogonal. In the four conditions, the alternative solution is less rigid than the simulated plane, but in conditions A-immobile and B-active, the alternative solution is more stationary.

The task for the observer is still to judge the orientation of the plane. The responses are said non-rigid if the orientation reported is nearer the alternative solution than the simulated solution.

The left graph in figure 4 shows the fraction of non-rigid responses for each four experimental conditions. We can see that in conditions A-immobile and B-active, the percentage of non-rigid responses is greater than 50%.

Model analysis: The right graph in figure 4 shows the results of the model in the same four conditions. We can see that, in accordance with the experimental results, the probability of non-rigid response is greater than 50% in both conditions A-immobile and B-active and smaller than 50% in conditions B-immobile and A-active.

This result is explained by the product between the prior on the absolute motion and the probability of the presented optic flow given the orientation and the relative motion. For the alternative response, the value of $P(\phi \mid \Theta\, \mathbf{X}_R)$

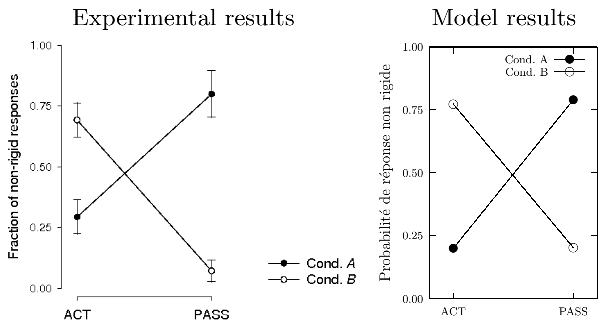

Fig. 4. Comparison of the model results with the experimental results.

is smaller than for rigid response. However, this is compensated for by a larger probability of the absolute motion $P(\mathbf{X}_A)$.

This example shows that a Bayesian model can reproduce results from psychophysical experiments.

4 Playing to train your video game avatar

Today's video games feature synthetic characters involved in complex interactions with human players. A synthetic character may have one of many different roles: a tactical enemy, a partner for the human, a strategic opponent, a simple unit among many, or a substitute for the player.

In all of these cases, the game developer's ultimate objective is for the synthetic character to act as if it were controlled by a human player. This implies the illusion of spatial reasoning, memory, common-sense reasoning, using goals, tactics, planning, communication and coordination, adaptation, unpredictability... In current commercial games, basic gesture and motion behaviours are generally satisfactory. More complex behaviours usually look much less lifelike. Sequencing elementary behaviours is a difficult problem, as compromises must be made between too-systematic ordering that always appears as an automaton and too-hectic behaviour that looks ridiculous.

We address this problem of real-time reactive selection of elementary behaviours for an agent playing a first-person shooter (FPS) game called *Unreal Tournament*. In this kind of game, a group of people play together via the Internet. Each of them can control a virtual avatar. This avatar may act and navigate in a virtual 3D environment. It may also interact with the avatars of other players or with autonomous characters called *bots* controlled by a program. In FPS games, the main interaction with the other players and bots consists of trying to slaughter them while surviving as long as possible.

The goal of this work is to demonstrate how, by using the Bayesian *inverse programming* technique, a player of a video game can teach an avatar how to play autonomously (see Fig. 5).

Fig. 5. Playing *Unreal Tournament* to train your avatar

4.1 Bayesian model

Variables: Six elementary reactive behaviours have been programmed based on these actions: attack foes (*attack*), search for weapon bonuses (*searchweapon*), search for health bonuses (*searchhealth*), explore the environment (*explore*), flee (*flee*), and detect danger (*detectdanger*). The goal of the program is to decide every tenth of a second which of these 6 behaviors should be applied. The game provide 7 "sensory" pieces of information to do so. Consequently, the relevant variables are as follows:

- B^t, the reactive behaviour of the bot at time t, which can take any of the six values: $\{attack, searchweapon, searchhealth, explore, flee, detectdanger\}$.
- H^t, the bot's health level at time t.
- W^t, the weapon that the bot is deploying at time t.
- OW^t, the opponent's weapon at time t.
- N^t, a Boolean variable indicating whether a noise has been heard recently.
- NO^t, the number of close enemies at time t.
- WP^t, the proximity of a weapon bonus.
- HP^t, the proximity of a health bonus.

Each of these 8 variables has to be considered for each instant between time 0 and T. This leads to a conjunction of $8 \times (T+1)$ variables[4]: $B^{0:t} \wedge \ldots \wedge HP^{0:t}$.

Decomposition: Usually, bots' behaviour in video games are specified using a scripting language with a set of rules use to sequence the activation of elementary reactive behaviours. Each rule is a list of conditions and constraints on the sensory variables that have to be verified to select the corresponding reactive behaviour. The generic form of such a rule is: if $condition_1$ and $condition_2$ and ... and $condition_n$ then $reactivebehaviour_i$.

There are 2 main ideas in inverse programming:

[4] The notation $B^{0:t}$ stands for the conjunction of $T+1$ variables: $B^0 \wedge B^1 \wedge \ldots \wedge B^T$

1. The first one is that knowing which reactive behaviour the robot is doing provides a lot of information on its' sensory variables.
2. The second is that knowing the reactive behaviour we can consider, as a first approximation, that these sensory variables are independent from one another.

Consequently, the rules are replaced by probability distributions expressing chunk of knowledge of the form if $reactivebehaviour_i$ then (approximately) $condition_j$. This is why it is called *inverse* programming.

The joint distribution is decomposed as follows.

$$
\begin{aligned}
& P(B^{0:t} \wedge H^{0:t} \wedge W^{0:t} \wedge OW^{0:t} \wedge N^{0:t} \wedge NO^{0:t} \wedge WP^{0:t} \wedge HP^{0:t}) \\
& = \prod_{t=1}^{T} \begin{bmatrix} P(B^t \mid B^{t-1}) \\ \times P(H^t \mid B^t) \times P(W^t \mid B^t) \times P(OW^t \mid B^t) \times P(N^t \mid B^t) \\ \times P(NO^t \mid B^t) \times P(WP^t \mid B^t) \times P(HP^t \mid B^t) \end{bmatrix} \\
& \times P(B^0 \wedge H^0 \wedge W^0 \wedge OW^0 \wedge N^0 \wedge NO^0 \wedge WP^0 \wedge HP^0)
\end{aligned}
\tag{6}
$$

In this decomposition, we assume that:

- The behaviour B^t at time t depends on the behaviour B^{t-1} at time $t-1$;
- The seven sensory variables may be considered to be independent of one another and independent of the past, knowing the behaviour B^t.

This decomposition is similar to the naive Bayes sensor fusion schem. The difference is that the fusion is usually done on some phenomenon whereas in our case the aim is an appropriate action to be executed.

Parametric forms: The eight terms of the decomposition $P(B^t \mid B^{t-1}) \cdots P(HP^t \mid B^t)$ are defined using tables specifying their discrete values. For instance, $P(H^t \mid B^t)$, defined in Table 1, gives the probability distribution for H^t (the bot's health level), knowing the behaviour B^t. We read the first column this way: given that the bot is in state *attack*, we state that it has a very low probability (10^{-3}) of having a low (poor) health level, a medium probability (10^{-1}) of having a medium (fair) health level, and a strong probability (0.899) of having a high (good) health level.

	attack	*searchweapon*	*searchhealth*	*explore*	*flee*	*detectdanger*
Low	10^{-3}	10^{-1}	x	10^{-1}	0.7	10^{-1}
Medium	10^{-1}	x	10^{-2}	x	0.2	x
High	x	x	10^{-3}	x	0.1	x

Table 1. $P(H^t \mid B^t)$

Learning: The value of these 8 tables may be either specified by the game designer (as in the example of Table 1) or, more interestingly, learned by observing a player.

The difficulty to learn this table from the data recorded by observing the player is that the only observable variables are the 7 sensory ones. The behavior selected by the player is not known (the value of B^t is missing). The very well known Baum-Welch [Baum, 1972, Rabiner, 1989] algorithm, a special case of the Expectation-Maximisation (EM) class of algorithms [Dempster et al., 1977], has been designed especially for learning when some data are missing. We apply an incremental version of the Baum-Welch algorithm as described by Florez-Larrahondo [2005]. In this algorithm, contrary to the classical Baum-Welch algorithm, the model parameters are re-estimated after each new observation. This algorithm can treat at least 10 sensori-motor acquisition by second allowing to do the learning on line as the player is using the game.

Question and inference: When playing autonomously, every tenth of a second, our bot must make a decision on its behaviour. It must answer the following probabilistic question:

$$P(B^t \mid b^{t-1} \wedge h^t \wedge w^t \wedge ow^t \wedge n^t \wedge no^t \wedge wp^t \wedge hp^t) \tag{7}$$

What is the probability distribution on behaviour at time t (B^t) knowing the behaviour at time $t-1$ (b^{t-1}) and knowing all the sensory information at time t ($h^t, \ldots, hp^t$)?

This question leads to a probability distribution, from which we draw a value to decide the actual new behaviour. The answer to this question may be easily computed as it is proportional to the product of the individual terms:

$$\begin{aligned} & P(B^t \mid b^{t-1} \wedge h^t \wedge w^t \wedge ow^t \wedge n^t \wedge no^t \wedge wp^t \wedge hp^t) \\ \propto\; & P(B^t \mid b^{t-1}) \times P(h^t \mid B^t) \times P(w^t \mid B^t) \times P(ow^t \mid B^t) \\ & \times P(n^t \mid B^t) \times P(no^t \mid B^t) \times P(wp^t \mid B^t) \times P(hp^t \mid B^t) \end{aligned} \tag{8}$$

Bayesian Program: The corresponding Bayesian program summarizing all this is then given by Fig. 6.

4.2 Results

The bots obtained this way after 15 minutes of online learning were faced to other bots or humans in tournaments. For instance, we collected data on series of ten tournaments with the native bots of the game (the one used in the commercial game to fight the human players). Our trained bots compare well with these bots, with skills corresponding to an average human player (see Table 2).

A much more detail description of this work and analysis of the obtained results may be found in Ronan Le Hy's PhD dissertation and publications [Le Hy, 2007, Le Hy et al., 2004]

$$
\text{Program}\left\{\begin{array}{l}
\text{Description}\left\{\begin{array}{l}
\text{Specification}\left\{\begin{array}{l}
\text{Relevant Variables:}\\
\quad B^{0:t},\, H^{0:t},\, W^{0:t},\, OW^{0:t},\, N^{0:t},\, NO^{0:t},\, WP^{0:t} \text{ and } HP^{0:t}\\
\text{Decomposition:}\\
P(B^{0:t} \wedge H^{0:t} \wedge W^{0:t} \wedge OW^{0:t} \wedge N^{0:t} \wedge NO^{0:t} \wedge WP^{0:t} \wedge HP^{0:t}) =\\
\quad \prod_{t=1}^{T} \left[\begin{array}{l} P(B^t \mid B^{t-1})\\ \times P(H^t \mid B^t) \times P(W^t \mid B^t) \times P(OW^t \mid B^t) \times P(N^t \mid B^t)\\ \times P(NO^t \mid B^t) \times P(WP^t \mid B^t) \times P(HP^t \mid B^t) \end{array}\right]\\
\quad \times P(B^0 \wedge H^0 \wedge W^0 \wedge OW^0 \wedge N^0 \wedge NO^0 \wedge WP^0 \wedge HP^0)\\
\text{Parametric Forms:}\\
\quad \text{All distributions are tables.}
\end{array}\right.\\
\text{Identification:}\\
\quad \text{In real time observing a player}
\end{array}\right.\\
\text{Question:}\\
\quad P(B^t \mid b^{t-1} \wedge h^t \wedge w^t \wedge ow^t \wedge n^t \wedge no^t \wedge wp^t \wedge hp^t)
\end{array}\right.
$$

Fig. 6. Sequencing a bot's reactive behaviours by inverse programming

Behaviour	Score
Random behaviour	43.2
Unreal Tournament bot (3/8) ($\simeq$ average human)	11.0
Incremental Baum-Welsh	8.5

Table 2. Performance comparison (3). Lower is better: minimum 0, maximum 100.

5 Conclusion

Numerous other models and robotics applications have been developed using the Bayesian Programming approach and the API ProBT®. Indeed, 14 PhD theses have been defended in different European Universities using this approach and formalism. Beside the 2 already presented subjects these theses cover the following fields: Bayesian Robots Programming [Lebeltel et al., 2004], Bayesian navigation on sensory–motor trajectories [Pradalier et al., 2005], Bayesian occupancy filters [Coué et al., 2005], Topological SLAM [Tapus, 2005], Bayesian CAD system [Mekhnacha et al., 2001], Computer-assisted surgery for total hip replacement (THR) [Amavizca-Ruiz, 2005], Probabilistic robotics arm control [Garcia-Ramez, 2003], Probabilistic contextual situation analysis [Ramel and Siegwart, 2004], Bayesian maps [Diard et al., 2005], Bayesian approach to action selection and attention focusing [Koike, 2005], Bayesian modelling of visual–vestibular interactions [Laurens and Droulez, 2007], Early speech acquisition [Serkhane et al., 2005]. These works prove the generality of the approach and the power of expression of the formalism. For more information and up to date panorama of on going developments please refer to Bayesian Programming site (www.bayesian-programming.org).

References

M. Amavizca-Ruiz. *3D human hip volume reconstruction with incomplete multimodal medical images. Application to computer assisted surgery of total hip prosthesis (THP).* PhD thesis, Institut National Polytechnique de Grenoble (INPG)., Grenoble, France, 2005.

L. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 3: 1–8, 1972.

F. Colas, J. Droulez, M. Wexler, and P. Bessière. A unified probabilistic model of the perception of three-dimensional structure from optic flow. *Biological Cybernetics*, in press, 2007.

C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessière. Bayesian occupency filtering for multi-target tracking. *International Journal for Robotics Research (IJRR)*, 25(1), 2005.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc., Ser. B*, 39:1–38, 1977.

J. Diard, P. Bessière, and E. Mazer. Merging probabilistic models of navigation: the bayesian map and the superposition operator. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS05)*, pages 668–673, 2005.

G. Florez-Larrahondo. *Incremental learning of discrete Hidden Markov Models*. PhD thesis, Mississipi State University, 2005.

B. J. Frey. *Graphical Models for Machine Learning and Digital Communication*. MIT Press, 1998.

R. Garcia-Ramez. *Programmation bayésienne des bras manipulateur*. PhD thesis, Institut National Polytechnique de Grenoble (INPG), Grenoble, France, 2003.

H. Helmholtz, von. *Handbuch der Physiologischen Optik.* Hamburg: Voss, 1867.

E. T. Jaynes. *Probability Theory: the Logic of Science*. Cambridge University Press, 2003.

J. Koenderik. Optic flow. *Vision Research*, 26(1):161–179, 1986.

C. Koike. *Bayesian Approach to Action Selection and Attention Focusing. Application in Autonomous Robot Programming*. Phd thesis, Inst. Nat. Polytechnique de Grenoble, Grenoble (FR), 2005.

J. Laurens and J. Droulez. Bayesian processing of vestibular information. *Biol Cybern*, 96(4):389–404, 2007.

R. Le Hy. *Programmation Bayésienne de Personnages de Jeux Vidéos*. Phd thesis, Institut National Polytechnique de Grenoble, Grenoble, France, 2007.

R. Le Hy, A. Arrigoni, P. Bessière, and O. Lebeltel. Teaching bayesian behaviors to video game characters. *Robotics and Autonomous Systems*, 47: 177 – 185, 2004.

O. Lebeltel, P. Bessière, J. Diard, and E. Mazer. Bayesian robot programming. *Advanced Robotics*, 16(1):49–79, 2004.

H. Longuet-Higgins. The visual ambiguity of a moving plane. *Proceedings of the Royal Society of London (B, Biological Sciences)*, 223:165–175, 1984.

K. Mekhnacha, E. Mazer, and P. Bessière. The design and implementation of a bayesian CAD modeler for robotic applications. *Advanced Robotics*, 15 (1):45–69, 2001.

J. Pearl. *Probabilistic reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

C. Pradalier, J. Hermosillo, C. Koike, C. Braillon, P. Bessière, and C. Laugier. The cycab: a car-like robot navigating autonomously and safely among pedestrians. *Robotic and Autonomous Systems*, 2005.

L. Rabiner. A tutorial on hidden markov models and selected applications inspeech recognition. In IEEE, editor, *Poceedings of the IEEE*, volume 77(2), pages 257–295, 1989.

G. Ramel and R. Siegwart. Multi-modal interaction using bayesian network to manage ambiguous situations. In *Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, LA, USA*, 2004.

J. A. Robinson. *Logic: Form and Function*. North-Holland, 1979.

J. Serkhane, J.-L. Schwartz, and P. Bessière. Building a talking baby robot. *Interaction Studies*, 2005.

A. Tapus. *Topological SLAM - Simultaneous Localization and Mapping with Fingerprints of Places*. Ph.d thesis, Ecole Politechnique Federale de Lausanne (EPFL), Lausanne, Switzerland, 2005.

S. Ullman. *The interpretation of visual motion.* Cambridge: MIT Press, 1979.

H. Wallach and D. O'Connell. The kinetic depth effect. *Journal of Experimental Psychology*, 45:205–217, 1953.

M. Wexler. Voluntary head movement and allocentric perception of space. *Psychological Science*, 14:340–346, 2003.

M. Wexler, I. Lamouret, and J. Droulez. The stationarity hypothesis: an allocentric criterion in visual perception. *Vision Research*, 41:3023–3037, 2001.