ProBT®

An API for Bayesian Programming

Pierre Bessière
Pierre.Bessiere@probayes.com

probayes.com

bayesian-programming.org

Mastering Uncertainty

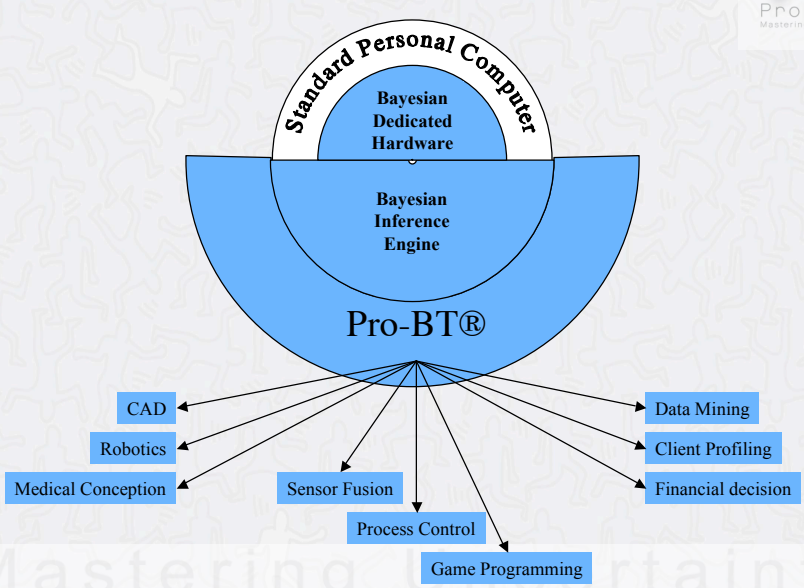
Company Vision

- Incompleteness: inescapable difficulty of complex problem
- Uncertainty: unavoidable consequence of incompleteness

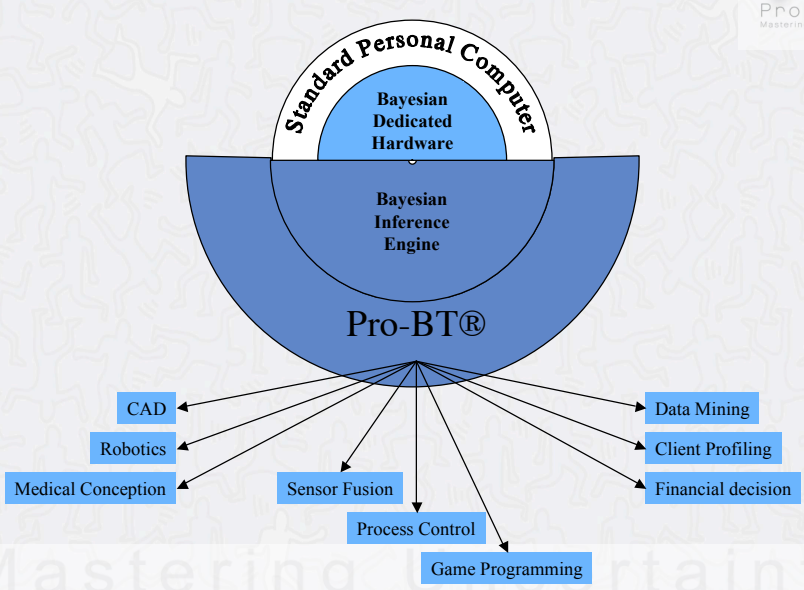
Bayesian Probability:
an alternative to Logic

Mastering Uncertainty

Bayesian Computer



ProBT® (1)



• Lebeltel et al. 2003, Autonomous robots

Bayesian Programming

Program

De…tion

Specification

– Variables

$S^0, \ldots\ldots, S^t, O^0, \ldots\ldots, O^t$

– Decomposition (Conditional Independance Hypothesis)

$$\mathbf{P}(S^0 \wedge \ldots\ldots \wedge S^t \wedge O^0 \wedge \ldots\ldots \wedge O^t) = \mathbf{P}(S^0) \times \mathbf{P}(O^0 \mid S^0) \times \prod_{i=2}^{t} [\mathbf{P}(S^i \mid S^{i-1}) \times \mathbf{P}(O^i \mid S^i)]$$

– Parametric Forms
or Bayesian Subroutines

$$\mathbf{P}(S^0) \equiv \mathbf{G}(S^0, \mu, \sigma)$$
$$\mathbf{P}(S^i \mid S^{i-1}) \equiv \mathbf{G}(S^i, A \bullet S^{i-1}, Q)$$
$$\mathbf{P}(O^i \mid S^i) \equiv \mathbf{G}(O^i, H \bullet S^i, R)$$

Identification

– Learning from instances

Question

$\mathbf{P}(S^t \mid O^0 \wedge \ldots\ldots \wedge O^t)$

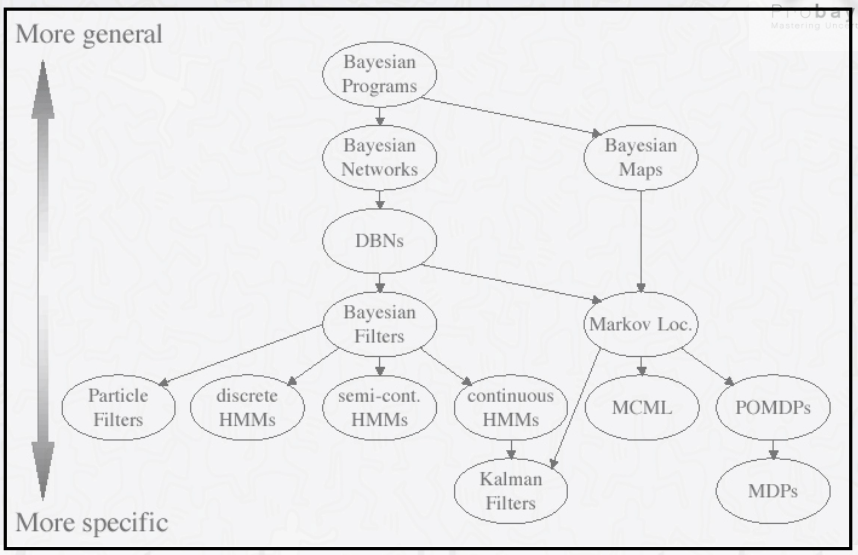
Mastering Uncertainty

Bayesian Programming (2)

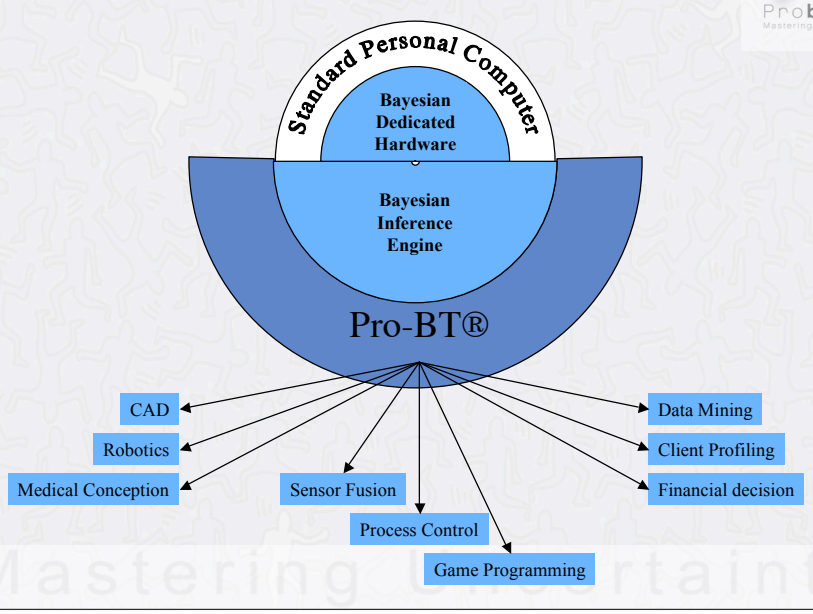
- Calling Bayesian Subroutines
- Bayesian Program if-then-else
- Bayesian Program Sequencing
- Bayesian Program Iteration
- Using Functions

Mastering Uncertainty

Related Approaches



ProBT® (1)



ProBT® (2)

```
main ()
{
  //Variables
  plFloat read_time;
  plIntegerType id("id",id_type);
  plFloat times[5] = {1,2,3,5,10};
  plSparseType time_type(5,times);
  plSymbol id("id",id_type);
  plSymbol time("time",time_type);

  //Parametrical forms
  //Construction of P(id)
  plProbValue id_dist[5] = {0.5,0.2,0.1,0.1,0.1};
  plProbTable P_id(id,id_dist);

  //Construction of P(time | id = john)
  plProbValue t_john_dist[5] =
    {20,30,10,5,2};
  plProbTable P_t_john(time,t_john_dist);
  //Construction of P(time | id = bill)
  plProbValue t_bill_dist[5] =
    {2,6,10,40,20};
  plProbTable P_t_bill(time,t_bill_dist);
```

```
//Construction de P(time | id)
  plKernelTable Pt_id(time,id);
  plValues t_and_id(time^id);
  t_and_id[id] = 0;
  Pt_id.push(P_t_bill,t_and_id);
  t_and_id[id] = 1;
  Pt_id.push(P_t_bill,t_and_id);

  //Decomposition
  // P(time id) = P(id) P(time | id)
  plJointDistribution
    jd(time^id,P_id*Pt_id);

  //Question
  //Asking the question P(id | time)
  plCndKernel Pid_t;
  jd.ask(Pid_t,id,time);

  //Read a time from the key board
  cout<<"P(id,time)= "<<Pid_t<<"\n";
  cout<<"Time? : ";
  cin>>read_time;

  //Getting P(id | time = read_time)
  plKernel Pid_readTime;
```

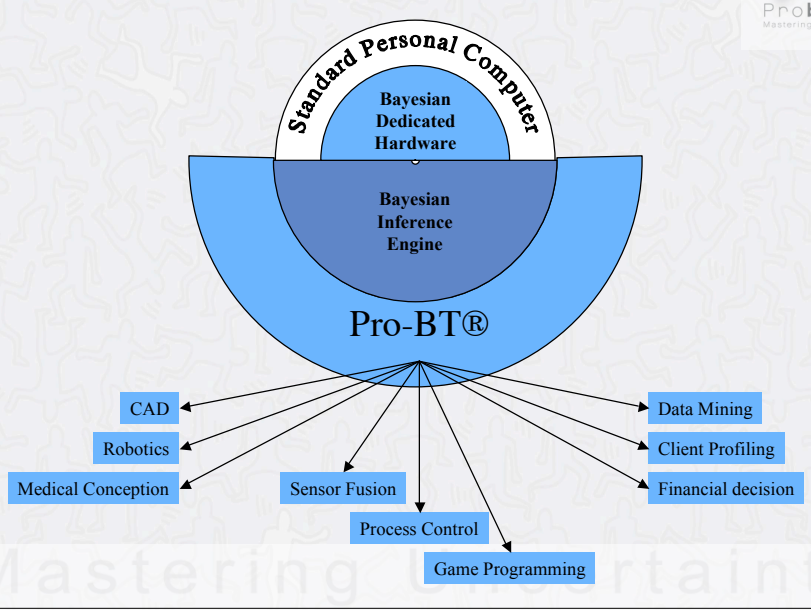
In C++, running on the following systems :

- Windows
- Unix
- Linux
- Mac

Interpreted versions (under development) :

- ImaLab (Collaboration with PRIMA)
- Python

Bayesian Inference Engine (1)



Inference Rules

$$P(X \wedge Y \mid \pi) = P(X \mid \pi) \times P(Y \mid X \wedge \pi)$$
$$= P(Y \mid \pi) \times P(X \mid Y \wedge \pi)$$

$$\sum_X P(X \mid \pi) = 1$$

$$\sum_X P(X \wedge Y \mid \pi) = P(Y \mid \pi)$$

Question

$$P(X^1 \wedge X^2 \wedge \ldots \wedge X^n) = P(L^1) \times P(L^2 \mid R^2) \times \ldots \times P(L^k \mid R^k)$$

$$P(Search \mid Known)$$

$$= \sum_{Free} P(Search \wedge Free \mid Known)$$

$$= \sum_{Free} \frac{P(Search \wedge Free \wedge Known)}{P(Known)}$$

$$= \frac{1}{Z} \times \sum_{Free} P(Search \wedge Free \wedge Known)$$

$$= \frac{1}{Z} \times \sum_{Free} P(L^1) \times P(L^2 \mid R^2) \times \ldots \times P(L^k \mid R^k)$$

- Marginalisation (integration) in high-dimensional space
- Searching the modes in high-dimensional space

2 modes: exact or approximated

- Symbolic simplification
- Numeric computation

Symbolic simplification

$$P(Search \mid Known) = \frac{1}{Z} \times \sum_{Free} \left[\prod_{i=0}^{k} P\left(L^i \mid R^i\right) \right]$$

Factorization:

$$P(Search \mid Known) = \frac{1}{Z} \times \prod_{j \in J} P\left(L^j \mid R^j\right) \times \sum_{Free1} \left[\prod_{k \in K} P\left(L^k \mid R^k\right) \right]$$

Sum to 1:
[Bessière et al. 2003]

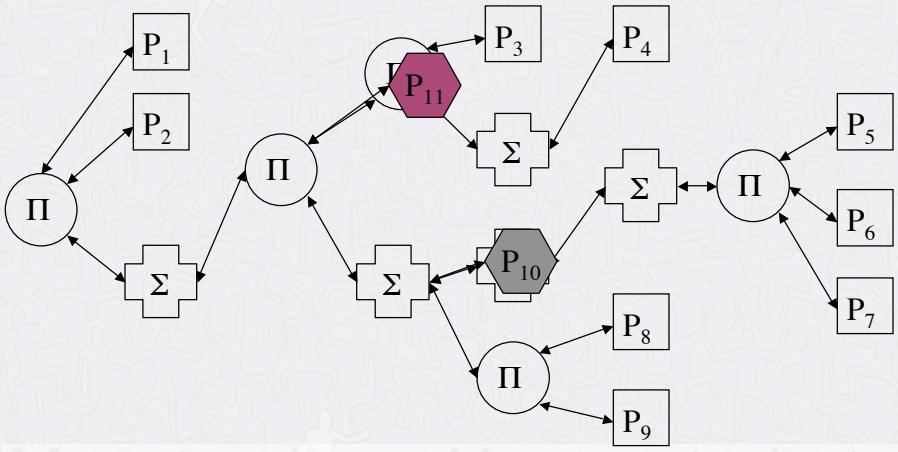
$$P(Search \mid Known) = \frac{1}{Z} \times \prod_{j \in J} P\left(L^j \mid R^j\right) \times \sum_{Free2} \left[\prod_{l \in L} P\left(L^l \mid R^l\right) \right]$$

Distributivity:
Generalized distributive law [Aji & McEliece2000]
Restrictions successives [Raoult & Smail2003]

$$P(Search \mid Known) = \frac{1}{Z} \times \prod_{j \in J} P\left(L^j \mid R^j\right) \times \sum_{Free3} \left[\prod_{m \in M} P\left(L^m \mid R^m\right) \times \sum_{Free4} \left[\prod_{n \in N} P\left(L^n \mid R^n\right) \times \ldots \times \sum_{FreeX} \left[\prod_{o \in O} P\left(L^o \mid R^o\right) \right] \right] \right]$$

Numeric computation (exact)

$$P(Search \mid Known) = \frac{1}{Z} \times \prod_{j \in J} P\left(L^j \mid R^j\right) \times \sum_{Free3} \left[\prod_{m \in M} P\left(L^m \mid R^m\right) \times \sum_{Free4} \left[\prod_{n \in N} P\left(L^n \mid R^n\right) \times \ldots \times \sum_{FreeX} \left[\prod_{o \in O} P\left(L^o \mid R^o\right) \right] \right] \right]$$



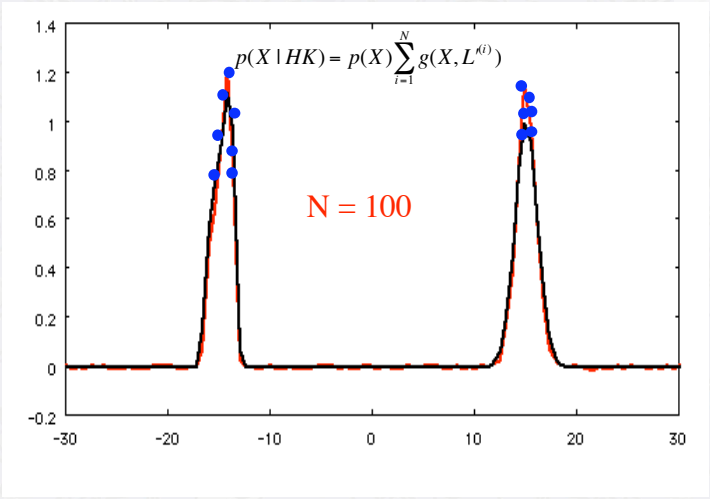
Arithmetic of very small numbers

- PlFloat a new type of number specific to encode probabilities
- 64 bits representation with 32 bits mantissa
- Very efficient implementation of the corresponding arithmetic

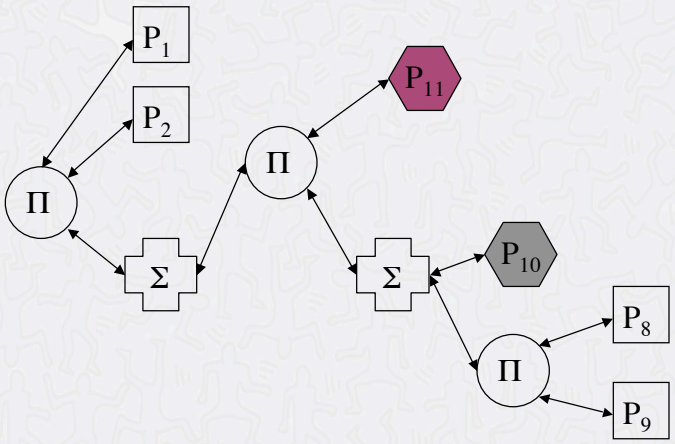
Numeric computation (approximated)

- MCMC for integration
- Genetic Algorithm to search the modes

MCMC + Annealing



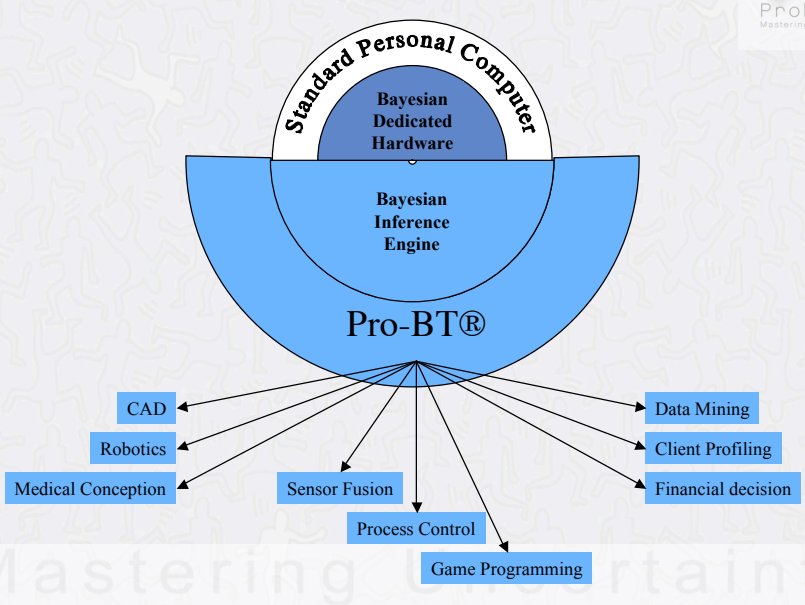
Multi-Resolution Binary Tree (MRBT)



Multi-Resolution Binary Tree (MRBT)

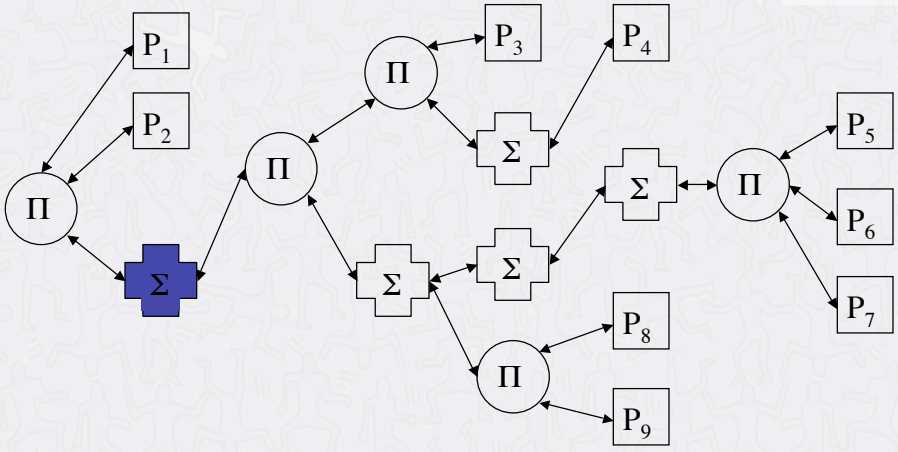
- Very compact representation
- Multi-resolution: more precision in high probability area
- Very efficient draw
- Incremental and anytime representation

Bayesian Hardware (1)

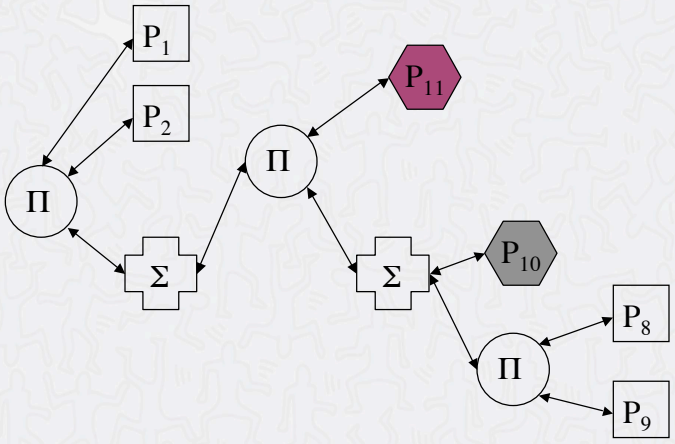


Bayesian Hardware (2)

Parallelization (1)



Parallelization (2)

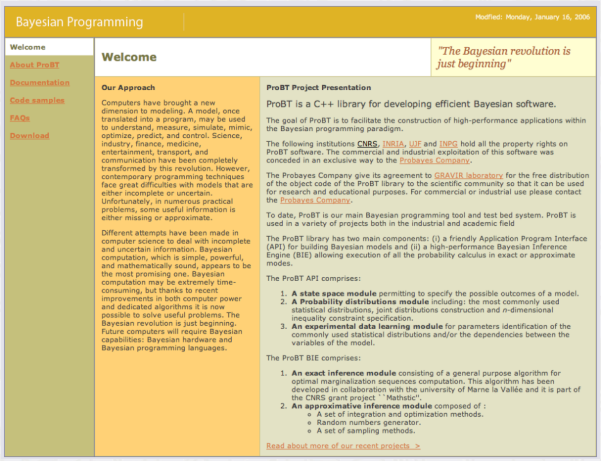


Historic

From fundamental research to industrial applications

- 1989 - Premières idées
- 1995 - Première version du moteur d'inférence (LISP)
- 1995 - Thèse sur l'incomplétude en robotique (E. Dedieu)
- 1997 - Deuxième version du moteur d'inférence (LISP)
- 1999 - Projet IMAG LAPLACE
- 1999 - Thèse sur la programmation bayésienne des robots autonomes (O. Lebeltel)
- 1999 - Thèse sur la CAO bayésienne (K. Mekhnacha)
- 2000 - Troisième version du moteur d'inférence
- 2001 - Transfert technologique - Incubation de la société
- 2001 - Projet Européen BIBA
- 2001 - Regroupement à GRAVIR au sein du projet SHARP
- 2002 - Disponibilité de Pro-BT (alpha)
- 2002 - Contrat SGAM
- 2002 - Projet RNTL télemaintenance
- 2003 - Thèse sur les cartes bayésiennes (J. Diard)
- 2003 - Thèse sur la programmation bayésienne des bras manipulateurs (R. Garcia)
- 2003 - Contrat SAMSE
- 2003 - Création de Pro-Bayes (septembre)
- 2003 - Thèse sur la fusion d'informations capteurs dans les systèmes d'assistance à la conduite automobile (C. Coué)
- 2004 - Livre « Bayesian Programming »
- 2004 - Thèse sur l'acquisition du langage chez le nourrisson (J. Serkhane)
- 2004 - Thèse sur le contrôle de véhicule (C. Pradalier)
- 2005 - Thèse sur la programmation bayésienne des robots autonomes (C. Koike)
- 2005 - Thèse sur la programmation et l'apprentissage bayésien des avatars dans les jeux vidéos (R. Lehy)
- 2005 - Thèse sur la perception de la forme par le mouvement (F. Colas)
- 2006 - Thèse sur la découverte automatisée de connaissances préalables (P. Danghautier)

Bayesian-Programming.org



ProBAYES recrute

Mastering Uncertainty