Bayesian Toolbox for Dynamic Distributed Decision Making

Vaclav Smidl

UTIA, Czech Academy of Sciences, Prague

10th September 2004
Outline

Motivation

Background
  Bayesian approach
  Distributed DM

Toolbox Description
  Models
  Communication
  Decision Making

Conclusion
Bayesian theory is a consistent, model-based approach to Decision Making (DM) under uncertainty.

- conceptually, it solves every problem,
- practically, analytical solutions only for simple problems,
- approximations => leads to different implementations

Traditional software tools work reliably for specific problems.
Bayesian theory is a consistent, model-based approach to Decision Making (DM) under uncertainty.

- conceptually, it solves every problem,
- practically, analytical solutions only for simple problems,
- approximations => leads to different implementations

Traditional software tools work reliably for specific problems.

The challenge: create a **general** software basis for Bayesian DM. (Including distributed decision making)
Classical “control” scenario:

Aim is to find the best possible action to reach desired behavior of the system. What is “the best”?
Aim is to find the best possible action to reach desired behaviour of the system. What is “the best”? 
Best is the action that leads to a minimum loss. Loss function:

\[ L = L(u, y, x, t). \]

How to deal with uncertainty?

- Ignore it!

\[ u = \arg \min_u L(u, y, \hat{x}, t). \]

- Model it!

\[ u = \arg \min_u \mathbb{E}_f(x) \{ L(u, y, x, t) \}. \]
Probabilistic Decision Making

Best is the action that leads to a minimum loss. Loss function:

\[ \mathcal{L} = \mathcal{L}(u, y, x, t). \]

How to deal with uncertainty?

Ignore it!

\[ u = \arg \min_{u^*} \mathcal{L}(u, y, \hat{x}, t). \]
Probabilistic Decision Making

Best is the action that leads to a minimum loss. Loss function:

\[ \mathcal{L} = \mathcal{L}(u, y, x, t). \]

How to deal with uncertainty?

Ignore it!

\[ u = \arg \min_{u^*} \mathcal{L}(u, y, \hat{x}, t). \]

Model it!

\[ u = \arg \min_{u^*} \mathbb{E}_{f(x)} \{ \mathcal{L}(u, y, x, t) \}. \]
Probabilistic Decision Making

Best is the action that leads to a minimum loss. Loss function:

$$\mathcal{L} = \mathcal{L}(u, y, x, t).$$

How to deal with uncertainty?

Ignore it!

$$u = \arg\min_{u^*} \mathcal{L}(u, y, \hat{x}, t).$$

Model it!

$$u = \arg\min_{u^*} \mathbb{E}_{f(x)} \{ \mathcal{L}(u, y, x, t) \}.$$

accurate: small number of measurements, model selection

expensive: analytical solution rarely available, approximations, heavy computations
Distributed Decision Making

How to control complicated large-scale systems? Large-scale complicated controls?
Distributed Decision Making

How to control complicated large-scale systems? Large-scale complicated controls?
Decentralized, multi-level control = distributed. Multiple simpler controllers.
How to control complicated large-scale systems? Large-scale complicated controls?
Decentralized, multi-level control = distributed. Multiple simpler controllers.
How to control complicated large-scale systems? Large-scale complicated controls? Decentralized, multi-level control = distributed. Multiple simpler controllers.

**Distributed Decision Making**
Examples of distributed Decision Making

- Colonies of robots: ROBO-FOOTBALL
- Process control (Rolling mill),
- multi-criteria decision making
Aim of the work

Create a software toolbox that:

- covers as many models as possible: crossroads, rolling mills, patients
- is flexible and extensible: code reuse, connectivity
Aim of the work

Create a software toolbox that:

- covers as many models as possible: crossroads, rolling mills, patients
- is flexible and extensible: code reuse, connectivity

LEGO
Aim of the work

Create a software toolbox that:

- covers as many models as possible: crossroads, rolling mills, patients
- is flexible and extensible: code reuse, connectivity

LEGO

Problems to overcome:

1. models of real world
2. optimization
3. communication
Aim of the work

Create a software toolbox that:

- covers as many models as possible: crossroads, rolling mills, patients
- is flexible and extensible: code reuse, connectivity

LEGO

Problems to overcome:

1. models of real world
2. optimization
3. communication

How to achieve maximum power with minimum code? Follow the theory...
Outline

Motivation

Background
  Bayesian approach
  Distributed DM

Toolbox Description
  Models
  Communication
  Decision Making

Conclusion
Generic Models

General model

\[ f(u, d, x) = \]

Breaking complex models into simpler ones
Generic Models

General model

\[ f(u, d, x) = f(u|d, x) f(d|x) f(x), \]

Breaking complex models into simpler ones => chain rule.

Bayesian networks:

- **Nodes**  probability density functions
- **Graphs**  connectivity (conditioning)

Bayesian Toolbox for Dynamic Distributed Decision Making
Generic Models

General model

\[ f(u, d, x) = f(u|d, x) \cdot f(d|x) \cdot f(x), \]

Breaking complex models into simpler ones => chain rule. Neglecting unimportant connections

Bayesian networks:

- **Nodes**: probability density functions
- **Graphs**: connectivity (conditioning)
Examples BN

Factor Analysis (Gaussian nodes)

Independent Component Analysis (non-Gaussian nodes)

Hidden Markov Model

HMM with mixture of Gaussian

Factor Analysis

Independent Component Analysis

Hidden Markov Model

HMM with mixture of Gaussian

Vaclav Smidl

Bayesian Toolbox for Dynamic Distributed Decision Making
Examples BN

Factor Analysis (Gaussian nodes)

Independent Component Analysis (non-Gaussian nodes)

Hidden Markov Model

HMM with mixture of Gaussian

General Inference methods - for Gaussians and discrete nodes.
BNT [Kevin Murphy]

Matlab Toolbox for inference of Bayesian networks.

+ Support for many pdf, Gaussian, Tables of discrete, Markov
+ Support for many inference schemes: MCMC, EM, J-trees,
+ Operations on graphs
  – Loose structure
  – Main purpose is testing and education
  – Does not support decision making and Communication

Attempt to build our own system with emphasis on decision making.
Communication Between Participants

What extensions we need to make

1 participant

Model: environment

Loss: individual
Communication Between Participants

What extensions we need to make

1 participant

Model: environment

Loss: individual

Communication is a dynamic process. When do we communicate? What to say? To whom?
Communication Between Participants

What extensions we need to make

1 participant

Model: environment

Loss: individual

Communication is a dynamic process. When do we communicate? What to say? To whom?

Communication is also Dynamic Decision Making.
Communication Between Participants

What extensions we need to make

1 participant

Model: environment

Loss: individual

Communication is a dynamic process. When do we communicate? What to say? To whom?

Communication is also Dynamic Decision Making.

Strategies: 1) selfish,
Communication Between Participants

What extensions we need to make

1 participant

Communication

Model: environment
Loss: individual

Communication is a dynamic process. When do we communicate? What to say? To whom?

Communication is also Dynamic Decision Making.

Strategies: 1) selfish, 2) hierarchical,
Communication Between Participants

What extensions we need to make

1 participant Communication

Model: environment + communication

Loss: individual

Communication is a dynamic process. When do we communicate?
What to say? To whom?

Communication is also Dynamic Decision Making.

Strategies: 1) selfish, 2) hierarchical, 3) cooperative,
Communication Between Participants

What extensions we need to make

1 participant

Communication

Model: enviroment

+ communication

Loss: individual

+group

Communication is a dynamic process. When do we communicate? What to say? To whom?

Communication is also Dynamic Decision Making.

Strategies: 1) selfish, 2) hierarchical, 3) cooperative,

Communication can be described probabilistically if we allow:

1) arbitrary marginalization, 2) pdf merging, 3) projections
Describing Loss Functions

Many different loss functions can be considered (and implemented in software). This leads to many optimization procedures for different models => combinatoric explosion. Can it be avoided?
Describing Loss Functions

Many different loss functions can be considered (and implemented in software).
This leads to many optimization procedures for different models => combinatoric explosion. Can it be avoided?
Define the target ideal behaviour by probability functions:

\[ f(u, y, x) \rightarrow ^I f(u, y, x), \]

The loss function is then a distance of the observed state of the system to the Ideal one:

\[ \mathcal{L}(u, y, x, t) = KL \left( f(u, y, x) \left\| ^I f(u, y, x) \right\| \right). \]
Describing Loss Functions

Many different loss functions can be considered (and implemented in software).
This leads to many optimization procedures for different models ⇒ combinatoric explosion. Can it be avoided?
Define the target **ideal** behaviour by probability functions:

\[ f(u, y, x) \rightarrow \text{if } f(u, y, x) \text{,} \]

The loss function is then a distance of the observed state of the system to the Ideal one:

\[ \mathcal{L}(u, y, x, t) = KL(f(u, y, x) \parallel \text{if } f(u, y, x)) \text{.} \]

Advantages:
- consistent with models,
- analytical solution is available (complicated)
- allow simple dynamic changes (communication)
Distributed Decision Making is a challenge for future research.
Can be formulated in Bayesian theory using known principles.
Setting up the framework, many pieces are to be filled.
We would like to manage the development as an open source project everyone is welcome to join
implemented in MATLAB, and ANSI C languages (any other language is welcome!).