Probabilistic graphical models for bayesian learning of state flow models in non stationary environments

Carlo Regazzoni
ISIP40 research group

ISIP40 research group is coordinated by Prof. Regazzoni and composed by:

- 1 Assistant Professor
- 8 Ph.D. students
- 7 Research grants
- 8 External collaborators

Main research topics:
- Cognitive Dynamic Systems
- Multisensor Signal Processing for Surveillance
- Software and Cognitive Radio
- Bayesian Object and Situation assessment
- Interactive and Cognitive Environments
Erasmus Mundus Joint Doctorate

- Coordinator of Erasmus Mundus Joint Doctorate on Interactive Cognitive Environments (ICE) (2010-2017) and JD ICE with Queen Mary University of London (2015-2019)
- 5 Universities
- Co-supervision, Mobility, shared syllabus/summer school, joint/double degrees
- Approx. 40 active students first four cohorts: main topics activity recognition, emotion analysis synthesis, networked embedded sensory infrastructures
- E-health, smart city, domotics, security applications fields
Outline

- **Introduction:** Scope of the talk

- **Background:**
  - **Cognitive Dynamic Systems:** Cognitive analytics, Interactions, Stability, Self awareness, Hierarchical representation, Cognitive Control, Global CDS goal
  - **Probabilistic Interaction-driven CDS models:**
    - Probabilistic Graphical models: Dynamic Bayesian Networks
    - Self awareness: DBN Representation, Loss of stability in DBNs, Inference in PGM
    - Cognitive control Reference model, PGM for incremental CC, Generative switching models

- **Learning of flow models:** state flow learning
  - Contextualized top down flow model learning: Hierarchical vocabularies, Techniques (SOM, NG etc.)
  - Flow models learning — Top down techniques, Flow models from sparse trajectory samples (Bayesian regression Gaussian Processes)
  - Switching variables and flow joint learning

- **Examples:** Non linear and linear flow models incremental learning

- **Conclusions**
This scope of the talk is:

- To define state flow models for single and interacting objects capable to represent self-awareness and to describe piecewise stationary situations.

- To define characteristics and examples of incremental learning techniques that can be used to learn such models.

- To contextualize presented techniques as Cognitive dynamic systems using Probabilistic Graphical Models where complex representation and inference can be jointly performed.

- To discuss conclusions and address open research applicative frameworks that can be based on above concepts.
Cognitive Dynamic Systems
Background: Cognitive Dynamic Systems

- Cognition is **embodied** into body of an organism **situated** in the structured environment from which receives sensory information [3].

- Understand the cognitive processes involves:
  - *Embodiment* concept: organism’s body composed by sensors and motor surface.
  - *Situatedness* concept: the structured environments in which the organism is immersed.
  - The individual organism’s *experiences*.
  - The neuronal process connected to the *behavior* of the organism.

- **Cognitive Dynamical Systems aim at providing tool** to develop a scientific theory of automatic cognition.

Application domain examples (1)

**Cognitive building**

SMART BUILDING
Cognitive Entity *PHYSICAL BODY*

- Actuators
- Sensors

Interaction between *Observed Entities*
Application domain examples (2)

Cognitive Drone

- EMBODIMENT of COGNITION
  - Cognition is situated in the physical body of the active entity
Background: Cognitive Dynamic Systems

- Working definition Cognitive Dynamic Systems (CDSs)

“Cognitive dynamic systems build up rules of behavior over time through learning from continuous experiential interactions with the environment, and thereby deal with environmental uncertainties”

Background: Cognitive Dynamic Systems

A Cognitive Dynamic System, following Simon Haykin in his attempt to give a systematic view of the wider field of Cognitive Artificial Systems in different application fields like, Radio, Radar, Video Surveillance, etc. can be described as:

1) a system whose design “closely mimics the human brain, be it visual, auditory, or some other sensory kind; cognitive radar, a remote sensory application, is a good example of this first class of cognitive dynamic systems”;

2) a system whose design “is motivated by human cognition; cognitive radio, a communication application, is a good example of this second class of cognitive dynamic systems”.

Cognitive analytics can be defined as the capability of a CDS to maintain dynamic stability conditions on the top of sensor (e.g., video) analytics by:

- detecting/correcting its own state;
- predicting/correcting future events/situations;
- learning from experiences new models at different abstraction levels.

Such an autonomous system (Cognitive Dynamic System) is supposed to be constituted by a “body” provided by sensors, actuators, and by a “brain” i.e., representation/processing hardware/software.
### Example Sensor analytics functionalities (Video)

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Level(s)</th>
<th>Extended Human Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multisensoriality</td>
<td>Signal</td>
<td>Spatial/multimodal observation</td>
</tr>
<tr>
<td>Video storage/retrieval</td>
<td>Signal/Feature</td>
<td>Past time/memory recall</td>
</tr>
<tr>
<td>Attention focusing (e.g. Video motion detection)</td>
<td>Signal/Feature</td>
<td>Observation Selection/Saliency</td>
</tr>
<tr>
<td>Tracking</td>
<td>Feature</td>
<td>Sub/Symbolic Temporal association and state estimation</td>
</tr>
<tr>
<td>Object recognition/identification</td>
<td>Feature</td>
<td>Observation tagging. Label-Associated information recall</td>
</tr>
<tr>
<td>Behavior analysis</td>
<td>Object/Event</td>
<td>Single object dynamic based temporal labeling</td>
</tr>
<tr>
<td>Situation awareness</td>
<td>Event/Situation</td>
<td>Multiple Object interaction based labeling</td>
</tr>
</tbody>
</table>
## Cognitive analytics

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Level(s)</th>
<th>Extended Human Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction</td>
<td>Signals to Situations in future temporal layer</td>
<td>Future time temporal anticipation</td>
</tr>
<tr>
<td>Self awareness</td>
<td>Signals to Situations</td>
<td>Internal system parameters self representation/reasoning</td>
</tr>
<tr>
<td>Dynamical adaptation</td>
<td>Current and future Situations</td>
<td>Representation/reasoning on relationships between internal/self system parameters and situation analysis to drive system actions</td>
</tr>
<tr>
<td>Learning</td>
<td>All levels</td>
<td>Updating/creating new models based on past experiences</td>
</tr>
</tbody>
</table>
Cognitive analytics - trend

Information and Space Unit per Time and Space Unit

Cognitive Abilities

- Signals
- Features
- Events
- Situations
- Prediction
- Self-Aware
- Dynamic Adaptation
Background: Cognitive Dynamic Systems

- **Interaction Driven- Cognitive Dynamical Systems (ID-CDS)** as Data Fusion architectures are based on observed and actively modulated **interactions** among the system self and multiple users.
- Their specificity is that they rely on a hierarchical representation that since lower levels is based on **coupled contextual representations** of:
  - **dynamic information** relevant to model interactive observed and proactive behaviors and to recognize them from observed sensory data
  - **dynamic inference and learning paradigms** and **processes** for efficiently translating hierarchical information into actions and **services** shared by the physical changes induced by the system onto the interacting world context

[5 Regazzoni et al SMC]
Example: a ID-CDS for surveillance

- An ICE aiming at maintenance of security level is supervised by an operator.
- Operator collects networked sensory data and takes decisions on status of networked sensors and actuators.
- Interactions analyzed:
  - Interactive behaviors of observed users in the scene (e.g., smart patrolling (guard + environment) and intrusion management behaviors (guard and thief)).
  - Multi-purpose associative representations of interactions between:
    - observed situation
    - status of operator controllable actuators (requires extended embodied cognition)
Example: a ID-CDS for surveillance

On line Context awareness enabled functionalities:
- Intelligent multi-sensor Distributed Data fusion
- Event detection
- Situation awareness
- Observed interactions evolution prediction

Active Interactions
- Distributed competitive/cooperative decision making
- Self representation for embodied cognition driven interactive situations management
- Associative/dispositional semantic representation of interactions allowing explicit causal reasoning

Example: ICE Security preserving system

ICE system

Multi sensors

Actuators

Cooperative entity

Non-cooperative entity

Environment
Observed interactions: examples

Observed people Interactions: situation awareness

Proactive Interactions

- Environmental control aim at control/influence users’ behaviors based on predictive/customized situation assessment. Interaction between sensed representation and human operator actions can be learned from experience.
Stability in Cognitive Dynamic Systems

- Dynamic System takes into account embodied and situated cognition by adaptively changing their state as a function of:
  - Stability (e.g. attractor states).
  - Loss of stability (e.g. bifurcations).
  - New stability (e.g. new attractor states).

- The goal of Dynamic System: to maintain stability of the equilibrium between the object and the environment (homeostasis, i.e. maintenance of the proper level of security and/or safety).
Examples: Cognitive environment with crowd

Observed group interactions: Crowd analysis

PETS-2009

Stability maintenance can be represented by sequences of contextual states.
Proactive interactions
Self awareness and Cognitive control in CDS
Goals for a ID-CDS: Self awareness

- Dynamic supervised stability maintenance has to be defined in the context of the “first person” of the CDS.

- Each observed Object and Situation has to be represented as an interaction between that external object and the dynamic “Self representation” of the CDS body.

- A subset of Issues is how to represent:
  - CDS Self
  - (Single) Objects and Situations dynamics as interactions between CDS Self and such objects
  - Interactions between objects and situations not apparently involving state changes of CDS itself
Neuroscience evidence for Interaction as a basis for Cognition

- A neuroscience viewpoint: *Cognition* and *interaction* are two inter-related functionalities of biological system.

- Cognitive reasoning is based on *dynamical dispositional representations* of the interactions between an object in contextual scenario and the changed organism state it causes [16].

- Such *dispositional representation of external objects wrt a self object* is the key aspect of the knowledge embedded on *bio-inspired ID-CDS*.

Damasio model: Somatic representations

- Damasio proposes a computational theory based on *somatic representations*.

- Somatic-representations leads to *somatic marker hypothesis* → logical strategies to decide among multiple available options, how various stimuli make the body react.

- Damasio explains that the representations in the mind are not stored as facsimiles of the original objects, but through the “*dispositional representations*”.
Goals for a ID-CDS: Self awareness

- Homeostasis in a complex system implies multiple levels of representations should be maintained.
- Hierarchical representation allows a CDS to separate the representation of the type of dispositional model with respect to the sequence of dispositional lower level states necessary to maintain a specific adaptive stability maintenance behavior.
- Iteration of the above separation process on multiple levels allows abstraction and coding in a rich complex environment with a large vocabulary.
Goals for a ID-CDS: Self awareness

- For example, a set of different behaviors and their sequence can be described at a higher level.
- Single behaviors can describe sequence of “normality” state dispositions when one of the available behaviors is fixed. i.e. situated behaviors in particular already experienced contextual labeled situations (here described by behavior type variable.
- So dynamic lower level models can well represent dispositional situated models of a single object.
Goals for a ID-CDS: Self awareness

- If behavior type class concept in the above example is extended, each discrete variable assuming finite values can be used as switching variable to be associated with a set of object behaviors.

- For example an object behavior type can be defined as a behavior type of another interacting object and vice versa, so defining a coupling among dispositional models.

- Coupled dynamic models of interacting objects can represent interactions:
  - between observed object and CDS states
  - between couples of observed entities (e.g. external entities, environment, etc.)
Goals for a ID-CDS: Self awareness

- Let us associate a dynamic level representation with a variable \( s(t) \) in a finite set. Each value of the variable can index a different homeostatic model \( h(X(t),s(t)) \) describing dispositions of lower level variable \( x(t) \) under a fixed \( s(t) \) value. Together, the associative couple \( (s(t),h(X(t),s(t))) \) represents:
  - Semantic of the behavior over a the time range of behavior type itself (e.g. a vehicle turning left and then going straight)
  - Expected dynamics of the represented object state \( x(t) \) during the evolution of a single behavior (similar to Fixed Action Pattern FAP in neuroscience) e.g. how car should move when turning left.

- The couple represents a basic unit for representing self awareness
Goals for a ID-CDS: Self awareness

- Self awareness can be described as the capability of a system to represent not only how it is sensing or doing, but also to distinguish the behavioral way from other possible behavior types.
- Self awareness is generally joined with context awareness.
- Coupling behavioral (dispositional) homogeneous models related to CDS state and external object state can be the basis to represent self awareness in a CDS system.
- For example coupled hierarchical Interaction models should describe variability of the behavioral steps of a CDS depending on contextual dynamic patterns of an external object (e.g. dynamic of a intelligent car on a road describing a car as CDS and road as external object)
- Each coupled model can describe contextualized normality of a given dynamic interaction wrt coupled object change of states
Hierarchical representation in Data Fusion systems

Example: Data Fusion systems can be provided of different level of inference associated with a hierarchical representation of estimated hidden state variables.

<table>
<thead>
<tr>
<th>Types of inference</th>
<th>Types of data/processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat analysis</td>
<td>Contextual analysis</td>
</tr>
<tr>
<td>Situation assessment</td>
<td>Analysis of Mission, goal, situation</td>
</tr>
<tr>
<td>Behavior of entity</td>
<td>Time and local geometric analysis</td>
</tr>
<tr>
<td>Identity</td>
<td>Parametric data</td>
</tr>
<tr>
<td>Position velocity</td>
<td>Multiple data for position estimation</td>
</tr>
<tr>
<td>Existence of an entity</td>
<td>Raw sensor data</td>
</tr>
</tbody>
</table>
Example: a ID-CDS for surveillance

• Multiple sensors and actuators are part of the body of an ICE: their sensing and actuation domains are represented in the ICE

• Networked perceptions/activity information is analyzed and organized into multiple abstraction levels

• Multilevel representations are learned from virtual/real experiences. Uncertainty and model representation is based on probabilistic, bio-inspired techniques.

• Competitive/cooperative multilevel representations are needed for hierarchical decentralized decision strategies to control actuators
Goals for a ID-CDS: Cognitive Control

- Cognitive Control can be described as the problem of a CDS of using experiences to optimize application of available models as well as to incrementally update self aware representations.

- Self aware representations aim at maintain homeostasis under contextualized labeled situation,

- Cognitive control steps should include functionalities related to
  - Selection of optimal available models /Temporal detection of time when all available normality models anymore sufficient to represent
  - Learning of new dispositional models at state level
  - Creation and coupling of new labels for new behavior types with learned behavioral models
Goals for a ID-CDS: Model Selection

- Hierarchical CC has to be carried out simultaneously at different representation levels.
- Selection of a new model in a available set of models correspond to find a control action weighting more configurations \((s(t), h(x(t),s(t)))\) where prediction is more in line with evidence (minimum uncertainty).
- This step can be represented as a CC unit controlling parallel perception units associated with different dispositional models.
Hierarchical CC Functional Models [5,6,7] Probabilistic reasoning units (Haykin&Fuster)

- **Hierarchical cognitive dynamic system.**
  - CP: cognitive perceptor;
  - CC: cognitive controller;
  - PRM: probabilistic reasoning machine.

- Each level can “observe” its lower and upper levels

Probabilistic reasoning units (Haykin & Fuster)

- Probabilistic reasoning unit: feedback mechanism for one layer in the cognitive dynamic system
Goals for a ID-CDS: stability maintenance

- Estimation of a disposition of $s(t)$ to change its value, in correspondence of a simultaneous disposition of a change of a dynamic behavioral model used for predicting next instant lower level state variable is the basis for selection.

- Dynamic detection of loss of stability can be described as simultaneous increasing of uncertainties of all in so far experienced models above a threshold.

- It corresponds to abnormality detection i.e. homeostasis violation, model dataset

- Loss of stability implies upper level to be updated in CC chain to allow top down reset of CP chain
The JDL model is improved since
- EACH hierarchy level is provided with a feedback loop
- The whole approach has a probabilistic foundation
Goals for a ID-CDS: new dispositional models

- Self aware representations
- Loss of stability auto detection triggering within Cognitive control process
- Incremental (or off line) experience based learning of new model after abnormality “survival”
- Creation of new symbols and new models to manage the new situation
- Closed loop control with feedback
- Need of suitable and coherent: multilevel representation methods, machine learning techniques, Cognitive control frameworks
Probabilistic Graphical Models for CDS Representation and Experience based Learning
PGM models for Somatic representations

- **Dispositional representations** should be capable to represent different probable ways that an object can follow when changing in time its state.
- Sensorial and temporal model uncertainty should be part of such representations to allow descriptions to be sufficiently general to capture stability within context variability.
- (Dynamic) Bayesian networks including time are perfect candidates to represent both self CDS and external objects.
Probabilistic graphical models

- Two major classes of graphical models:
  - Bayesian Networks (directed graphical models)
  - Markov random fields (undirected graphical models)

- Directed graphical models:
  - The links of the graphs have a particular directionality indicated by arrows.

- Undirected graphical models:
  - The links do not carry arrows and have no directional significance.
PROBABILISTIC TOOLS
A computational framework that allows the modeling of empathic human behaviors and interactions in a natural and sound way is the **Probabilistic graphical model framework** (PGMs)

- It provides a statistical framework to model interactions and cause-effect relationships
- It enables to formalize and handle the uncertainties

The basic idea is to provide a graphical tool to decompose a multivariate probability distribution into a factored form by providing an intuitive and manageable visual description

Graphical models can be divided into: Directed Acyclic Graphs (DAGs) and Undirected Graphs (UGs)

DAGs are preferred where it is relevant to model cause-effect relationships like in interaction analysis

---

Probabilistic graphical models

- **Definition:** Diagrammatic representations of probability distribution.

- **Properties:**
  - A simple way to visualize the structure of a probabilistic model
  - Insights into the properties of the model, including conditional independence properties
  - Expressing complex computations in terms of graphical manipulations
A graph comprises **nodes/ vertices**

Nodes are connected by **edges/ links/ arcs**

Each node represents one (or a group of) **random variable(s)**

The links express **probabilistic relationship** between these random variables
Probabilistic graphical models

- Two major classes of graphical models:
  - Bayesian Networks (directed graphical models)
  - Markov random fields (undirected graphical models)

- Directed graphical models:
  - *The links of the graphs have a particular directionality indicated by arrows.*

- Undirected graphical models:
  - *The links do not carry arrows and have no directional significance.*
Probabilistic graphical models

- Directed graphical models:
  - *Useful for expressing causal relationships between random variables.*

- Undirected graphical models:
  - *Better suited to express soft constraints between random variables.*
PGM reference class for CDS: Bayesian networks

Examples

Bayesian networks: hierarchy and semantics

Dynamic Bayesian Network (DBN): extension to handle temporal models

It is defined as a pair \((B_0, B \rightarrow)\) where \(B_0\) defines the prior \(p(v_0)\) and \(B \rightarrow\) is a two-slice temporal Bayes network defined by means of a DAG [17] as follows:

\[
p(v_k|v_{k-1}) = \prod_{i=1}^{N} p(v_i^k|Pa(v_i^k))
\]

Where \(v_i^k\) is a node at time \(k\) and \(Pa(v_i^k)\) are the parents of \(v_i^k\).

Unfolding the network with time….

Recursive Bayesian State estimation can be used to estimate the state of the system at each time step (Kalman Filter, HMMs, Monte Carlo methods, …)

doi: 10.1109/MSP.2010.938078
Dynamic Bayesian networks

- Dynamic Bayesian Networks (DBNs) can be considered as an extension of Bayesian Networks to handle temporal models.
- The term “dynamic” is due to the fact that they are used to represent a dynamic model (a model with a variable state over time).
- A DBN is defined by $(B_0, B \rightarrow)$ where $B_0$ defines the prior probability over the state and $B \rightarrow$ is a two-slice temporal Bayes net (2TBN) which defines how the systems evolves in time.
There are two types of edges (dependencies) that can be defined in a DBN. **Intra-slice** topology (within a slice) and **inter-slice** topology (between two slices).

Dynamic Bayesian networks

- The decision of **how to relate two variables**, if either intra-slice (aka intra-time-slice) or inter-slice (aka inter-time-slice) depends on how tight the coupling is between them.
- If the **effect** of one variable on the other is **immediate** (much shorter then the time granularity) the influence should manifest as **intra-slice** edge.
- If the **effect** is slightly **longer-term** the influence should manifest as **inter-slice** edge.
- An inter-slice edge connecting two instances of the same variable is called **persistence-edge**.
- Hidden Markov Models (HMMs) and Kalman Filter Model (KFM) are specific nontrivial examples of DBNs.
- They are formed by one hidden variable with persistence links between time steps and one observed.
Dynamic Bayesian networks

HMM

- HHM is characterized by one **discrete hidden node**.
- The probabilities that have to be defined are:
  - $p(x_0)$ that is the **initial state distribution** and represents the uncertainty on the initial value of the state.
  - $p(x_k | x_{k-1})$ that is the **transition model**. It describes how the state evolves in time.
  - $p(z_k | x_k)$ that is the **observation model** and represents how the observations are related and generated by the hidden state. It is also called **likelihood**.
Dynamic Bayesian networks
KF Models

- KFM is characterized by one **continuous hidden node**.
- All nodes are assumed to be **linear-Gaussian distributions**.
- The probabilities then defined as:

  - **Initial state**
    \[ p(x_0) = N(x_0, Q_0) \]

  - **Transition model**
    \[ p(x_k | x_{k-1}) = N(Fx_{k-1} + Gu_k, Q) \]

  - **Observation model**
    \[ p(z_k | x_k) = N(Hx, V) \]
PF is characterized by one **continuous hidden node**.

All nodes are assumed to be generic not parametric distributions.

The probabilities that have to be defined are always the same, but they are described though a set of particles.

An importance function (usually Gaussian) over the state space has to be defined depending on the PF model as reference for weighting particles.
Dynamic Bayesian networks
Data Fusion models

- There mainly three ways to fuse observations in DBNs

  - Conditionally independent fusion
  - Linearly conditionally dependent fusion
  - Conditionally dependent fusion
Dynamic Bayesian networks – Multisensor Data Fusion

- **Mathematically** this relations can be expressed, defining 
  \[ Z_k = \{ z^1_k, z^2_k, \ldots, z^L_k \} \]
  as the set of different observations (or sensors), as:

  - **Conditionally indepent fusion**
    \[ p(Z_k \mid x_k) = p(z^1_k \mid x_k) p(z^2_k \mid x_k) \ldots p(z^L_k \mid x_k) \]

  - **Linearly conditionally dependent fusion**
    \[ p(Z_k \mid x_k) = \alpha^1_k p(z^1_k \mid x_k) + \alpha^2_k p(z^2_k \mid x_k) + \ldots + \alpha^L_k p(z^L_k \mid x_k) \]
    Subject to: \[ \sum_{i} \alpha^i_k = 1 \]

  - **Conditionally dependent fusion**
    \[ p(Z_k \mid x_k) = p(z^L_k \mid z^{1:L-1}_k, x_k) p(z^{L-1}_k \mid z^{1:L-2}_k, x_k) \ldots p(z^1_k \mid x_k) \]
PGM models for Hierarchical self awareness

- DBN can generalize state estimation models onto multiple hierarchy levels.
- Dispositional knowledge can be represented by means of inter slice probability models.
- DBNs with only one layer can represent dispositional knowledge in a fully stationary domain.
- Further hierarchy has to be included in the DBN to allow piecewise stationarity to be represented.
- Higher level discrete random variables can be used to represent such set of contextualized stationarity models.
- Switching dynamical systems are the PGM class.
Switching dynamical systems

\[ r_t \sim \pi_{r_t, r_{t-1}} \]
\[ x_t \sim f_{r_t}(x_t | x_{t-1}, \theta_{r_t}) \]
\[ y_t \sim g_{r_t}(y_t | x_t, \phi_{r_t}) \]
\[ r_t \in \{1, \cdots, K\} \]
Dynamic Bayesian networks
Interaction oriented DBN structures

- Interactions among a couple of objects can be represented by a PGM where multiple DBN single object models are linked.
- Inter object / Inter slice links can represent the influence on one object state on other object state at next time instant and vice versa.
- Switching coupled DBN models can be a basis for representing homeostatic interactive behaviors.
- Depending on inter object level of links different classes of switching models can be defined.
Coupled Dynamic Bayesian Networks
Discrete coupling in ID-CDSs

Measurement on external world
Continuous level (low level)
Discrete level (Higher level semantic description)
Continuous level (low level)
Measurement on external world

Entity 1
Entity 2

State description (Random variable)

Temporal evolution
Interaction
Interaction
PGM models and loss of stability

- Both for single object and ID-DBNs, loss of stability can be represented as a consistent probability model change associated with a transition probability in the switching variable values.
- Switching variables can be described also as “super states”, that regulate and describe not synchronous semantic level transitions occurring when a different probabilistic (dynamic or observation) model has to be applied.
- Models for representing such discrete time variant transitions can be related to HMM (eg semi Markov processes) or Autobiograpical memories
Switching Model for multi-level analysis

- The dynamic evolution of the super states can be modeled by different transition models of an HMM: different models can be associated with each super state value.

- Corresponding to each value of super states a switch can be defined of a different dynamic state space model?

- For example, in a Switching Linear Dynamic System (SLDS). a discrete switch variable $s_t \in 1, \ldots, S$ can be used to select a different dynamic/observation state model.
  - The probabilistic model can be defined as follows:

  \[ p(y_{1:T}, x_{1:T}, s_{1:T}) = \prod_{t=1}^{T} p(y_t|x_t, s_t) p(x_t|x_{t-1}, s_t) p(s_t|s_{t-1}) \]

  - $s_t$ is a variable referred to super state label $l_t \in 1, \ldots, L$
Inference and Filtering in Switching Dynamical Systems

Filtering in Switching Dynamical systems Corresponds to calculation of joint belief:

\[ p(X_{0:k}, S_{1:k} | Z_{1:k}) \]

\( S_{1:k} = \{S_1 \ldots S_k\} \): sequence of discrete state

\( X_{0:k} = \{X_0 \ldots X_k\} \): sequence of continues state

\( Z_{1:k} = \{Z_1 \ldots Z_k\} \): sequence of observations

Dedicated Bayesian filtering methods should be applied to for solving this problem

- Markov Jump Particle Filter (MJPF)
  - Mixing Particle Filter and Kalman Filter
  - Works for linear and Gaussian case
- Rao-Blackwellized Particle Filter (RBPF)
  - Mixing HMM and Particle Filter
  - Works for non-linear and non-Gaussian case
PGM models for Cognitive Control

- Using Haykyn and Fuster model as reference one can associate two coupled switching models to CP and CC, similar to Coupled DBNs above.

- A difference is that observations for CC is dynamic uncertainty of CP and hidden states in CC are related to control switch modalities.

- An issue is how one can represent probabilistic reasoning unit within the PGM scheme.

- PRU jointly observes CP and CC uncertainty to select probabilistic control actions on CP.
Probabilistic reasoning units (Haykin&Fuster)

- **Probabilistic reasoning unit:** feedback mechanism for one layer in the cognitive dynamic system
In [Teddy] a first tentative to map PRU onto a DBN for controlling multiple trackers has been done

Another more integrated proposals can be found in [IEEE Trans Pietro]

This last approach uses Reinforcement learning as dynamical filter in CC and uncertainty of related actions to dynamically select how to integrate two KF filters for tracking

Add refs
PGM models for Incremental Cognitive Control

- However, a limitation of above approaches is that the switching label space in not incremental and new dynamic models cannot be learned
- How a PGM can be extended to be able to manage and learn non stationarities?
- A key issue is to be able to represent in a stationary way non stationarities, i.e. to add in the hierarchy an additional higher level representation level that has stable probabilistic properties and triggers multiple lower level modalities
- A technique is to use Dirichlet or similar approaches to design a PRU
Finite Mixture Model

- The distribution of data consists of sub-distributions each one is described by some parametric form

$$x \sim \sum_{k=1}^{K} \pi_k f(x, \theta_k)$$

Component parameters \(\{\theta_1, \cdots, \theta_K\}\)

Component probabilities \(\pi = [\pi_1, \cdots, \pi_K]\)
PGM for mixture model

\[ x_i \sim f(x, \theta_{z_i}), \ i = 1, \cdots, N \]

\[ z_i \sim \text{Cat}(\pi) \quad \pi = [\pi_1, \cdots, \pi_K] \]

often we have uncertainty about \( \{\pi_1, \cdots, \pi_K\} \) and \( \{\theta_1, \cdots, \theta_K\} \)
which is encoded in their prior probability

\[ \theta_k \sim g(\theta, \lambda), \ k = 1, \cdots, K \]

\[ \pi_1, \cdots, \pi_K \sim \text{Dir}(\alpha) \]
### Example of parametric distributions

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Definition</th>
<th>Notation</th>
<th>Parameters</th>
</tr>
</thead>
</table>
| Normal (Gaussian)             | $f(x | \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ | $x \sim \mathcal{N}(\mu, \sigma^2)$ | $\mu$ : mean  
                      |                              |           | $\sigma^2$ : variance   |
| Multivariate Gaussian         | $p_{x}(x_1, \ldots, x_k) = \frac{1}{\sqrt{(2\pi)^k | \Sigma|}} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right)$ | $x \sim \mathcal{N}(\mu, \Sigma)$ | $\mu$ : mean vector  
                      |                              |           | $\Sigma$ : Covariance matrix |
| Categorical                   | $f(x = i | p) = p_i$  
$x \in \{1, \ldots, k\}$                                                      | $x \sim \text{Cat}(p)$ | $ p = (p_1, \ldots, p_k)$  
                      |                              |           | Class probabilities |
| Uniform                       | $f(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b, \\ 0 & \text{for } x < a \text{ or } x > b \end{cases}$ | $x \sim \mathcal{U}(a, b)$ | $a$ and $b$            |
Mixture model example

\[ \theta_k = \{\mu_k, \Sigma_k\} \]
\[ \lambda = \{\mu_0, \kappa, \Psi, \nu\} \]

\[ \mu_k, \Sigma_k \sim \text{NIW}(\mu_0, \kappa, \Psi, \nu) \quad k = 1, \cdots, K \]
\[ x_i \sim \mathcal{N}(\mu_k, \Sigma_k), \quad i = 1, \cdots, N \]

\[ K = 4, \quad N = 1000 \]
\[ \alpha = 1, \quad \mu_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \kappa = 0.1, \quad \Psi = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix}, \quad \nu = 3 \]
Mixture Models – parameter Estimation

Estimating the set of parameter \( \{ \pi_1, \cdots, \pi_K \} \) and \( \{ \theta_1, \cdots, \theta_K \} \)

- **When we know number of components** \( K \)
  - EM algorithm, only efficient when components are Gaussian
  - MCMC, Gibbs sampling

- **When we do not know number of components** \( K \)
  - Guess \( K \) and use above methods
  - Use similarity-based clustering algorithm to find \( K \) (some times works)
  - Use Dirichlet process
Dirichlet Process Mixture

- Dirichlet process is used as prior distribution of mixture component’s parameter
- Dirichlet Process Mixture Model (DPMM)

\[ G \sim DP(\alpha, G_0) \]

\[ \theta_i \sim G \text{ for } i = 1, \cdots, N \]

\[ x_i \sim f(x, \theta_i), \ i = 1, \cdots, N \]

Problem:

Given \( x_1, \cdots, x_N \) estimate \( G \)
Dirichlet Process Mixture, alternative PGM

• It is not possible to generate sample from later model

• An alternative PGM for DPMM is constructed based on Stick-Breaking process (indicator variable)

\[ \pi \sim SBP(\alpha), \pi = \{\pi_1, \pi_2, \ldots\} \]

\[ z_i \sim Cat(\pi) \text{ for } i = 1, \ldots, N \]

\[ \theta_k \sim G_0 \text{ for } k = 1, 2, \ldots \]

\[ x_i \sim f(\theta_{z_i}) \text{ for } i = 1, \ldots, N \]
DPM and DBN

• HDP-HMM (Hierarchical Dirichelet Process Hidden Markov Model)
  • Now we can estimate number of hidden states
Switching models for Incremental Cognitive Control

- A switching label value can in this way be associated with
  - A category value i.e. a component
  - A parameter that can be associated with the pdf controlled by the switching component

- In the Dirichlet approach techniques like stick breaking can allow new components to be incrementally added in the learning phase, so making
  - Non stationary the set of components and its parameters
  - Stationary the particular process regulated by a specific component and its parameter
Incremental state flow models learning
Goals for a ID-CDS: Learning

- Different **vocabularies and syntax of switching variables implies** different performance of probabilistic DBN inference (prediction and update).
- Different vocabularies and syntax can correspond to **different semantic**.
- **Dynamic probabilistic models can be associated with “letters”**
- **Different machine learning techniques** learning jointly the vocabulary, their semantic and dynamic models can be defined
- **Stability definitions criteria** (and consequently loss of stability detection methods, i.e. switch detection) should be based on measures coming from observing the behavior of dynamic probabilistic filters running at a certain time instant.
Goals for a ID-CDS:
State flow learning in PGM CDS models

- If one concentrates in switching models related to dynamic model switching, what should be learned and from what data in a PGM model?

- \( p(x_t/x_{t-1}, s_t) \)
- \( p(s_t/s_{t-1}) \)

are respectively the sub semantic and semantic dispositional dynamical models to be learned. Values of st represent the “vocabulary”, xt represents the estimated system state.
Machine learning issues

- Nodes of the DBN: can a time variant not stationary DBN model be learned, where higher level nodes switching model are changed incrementally? (e.g. Dirichlet)
- New concepts (semantic) corresponding to new dynamic state changes should be learned (e.g. Techniques like SOM, ITM, or Neural Gas, etc.)
- As higher level discrete super states nodes in switching model PGM are associated with dynamic state models (observation or dynamic models) how can new flow model be represented and learned? (e.g. using Gaussian Processes to learn dynamic flow models from sparse samples), etc.
Learning superstates and their dynamic models

- Let us suppose we want just to learn off line

- $s_t$ i.e. the vocabulary of switching variables (superstates)
  AND

- $p(s_t/s_{t-1})$ where $s_t <> s_{t-1}$
  i.e. transition probabilities of events i.e. switching variables value transitions in time

- In this case one can later learn
  - Superstate conditioned flow model $p(x_t/x_{t-1},s_t)$
  - Event based conditioned flow models $p(x_t/x_{t-1},s_t,s_{t-1})$

- as well as versions of such models where superstates and events are related to different objects
Learning in switching models

The vocabulary of switching variables is essential to define which dispositional representations can be capable to represent how an object can change in time its state.

The state flow models are strictly coupled with each item of the vocabulary.

Within CDS using PGMs the choice of the vocabulary can be related to:

- Reducing state dimensionality
- Searching for a vocabulary that optimize (contextual) prediction
- Mixing two above aspects
Learning in switching models

- The optimal vocabulary should allow to code in an efficient way the sequence (semantic) as well as to prepare the self to predict next proactive and observed interactions.

- Techniques in machine learning that concentrated to learn switching variables vocabularies spam from methods:
  - more centered on semantics by learning nodes off line (e.g. ITM, SOM)
  - methods that do this incrementally (eg Neural Gas, Dynamic SOM, DPMM)
Hierarchical vocabularies
Super States and Events

- Switching variables can be described as components of categories that partition \textbf{State vector} space. Such components have been defined as superstates in []

- However, often considering as components at a higher level partitioning of state \textbf{variations, i.e.} distinguishing between state variations that belong to different classes, can be useful to discriminate among different dispositions/flow models

- Not every state variation is relevant, at semantic level

- \textbf{One can define events are relevant state changes as in [x]} and related CDS methods

- Events tells us when «something noticeable happens»

Super-states

**Issues:**
- What is relevant and what is not?
- How to obtain an event description?

If Events are defined as **Superstates variations**, than each couple of superstates can

**Superstates**
- Group many lower level states (clustering)
- Can be defined by hand, based on the semantics of the problem
- Can be learned by UNSUPERVISED LEARNING
- Can imply dimensionality reduction
- Are **discrete**
- Different superstate representation of course give **different ways of defining events**
Higher Level(s) in DBNs

\[ a_k \rightarrow a_{k+1} \]
\[ x_k \rightarrow x_{k+1} \]
\[ z_k \rightarrow z_{k+1} \]
Further reading

The self-organizing map (SOM) is mainly a method for **unsupervised learning**, based on a **grid of artificial neurons** whose **weights** are adapted to match **input vectors** in a **training set**.

It was first described by the Finnish professor Teuvo Kohonen and is thus sometimes referred to as a **Kohonen map**.

The SOM is one of the most popular neural computation methods in use, and several thousand scientific articles have been written about it. It is especially good at producing **visualizations of high-dimensional data**.
Self Organizing Map (SOM)

- The SOM is an unsupervised **Neural Network** technique that approximates an unlimited number of input data by a finite set of models arranged in a 2d (rarely 3d) grid, where neighbor nodes correspond to more similar models.
- The models are produced by a learning algorithm that automatically orders them on the **two-dimensional grid** along with their mutual similarity.
- A big variety of application fields
  - Unsupervised learning
  - Clustering
  - Classification
  - Data visualization
Other Self Organizing Networks (1)

- **G-SOM - Growing SOM** [2].
  - The map is allowed to have rows and column insertions if inputs are dense in a certain area.

- **GH-SOM – Growing Hierarchical SOM** [3].
  - Tries to reflect possible hierarchical structures in the input data.
Other Self Organizing Networks (2)

- NG – Neural Gas [4]
  - Neurons during the adaptation process distribute like a gas within the data space. **Topology preserving**.
- GNG – Growing Neural Gas (video) [5]
- H-GNG Hierarchical Growing Neural Gas [6]

Note: here input has the same dimension of the output.
Top down learning parametric dynamic models

- Top down techniques are available to learn parametric probabilistic models once components of a categorical distribution are fixed like:
  - Superstate conditioned flow model $p(x_t|x_{t-1}, s_t)$
  - Event based conditioned flow models $p(x_t|x_{t-1}, s_t, s_{t-1})$

- A more general formulation can be given as follows
Parameter learning in PGM - Parametric probability (1)

- CDFs are often of the form of some parametric functions
- Learning CDFs is equivalent to estimating the parameter of the function
- The parameters might be function of condition variable

\[
p(x|y) = f(x, y, \theta)
\]
\[
p(x|y) = g(x, \phi(y))
\]

Fix parameter, CDF is function of condition variable y

CDF is independent of y but parameter is function of y
Example: Joint offline HMM parameter learning

- Hidden Markov Model (HMM) is a special type of DBN with one observed variable and one hidden variable in each time slice.

  \[ x(t-1) \rightarrow x(t) \rightarrow x(t+1) \]

  \[ y(t-1) \rightarrow y(t) \rightarrow y(t+1) \]

  \[ x \text{ is hidden state variable, } x \text{ is discrete or categorical variable} \]

  \[ y \text{ is observed variable, } y \text{ can be discrete or continuous variable} \]

  Two CPDs:

  - State transition model: \( p(x(t)|x(t-1)) \)

  - State emission model: \( p(y(t)|x(t)) \)
HMM parameters

\[ x(t) \in \{1, \cdots, K\} \quad \text{and} \quad y(t) \in \mathcal{Y} \]

state transition model \( p(x(t)|x(t-1)) \) is a conditional categorical distribution
\[ x(t)|x(t-1) = i \sim \text{Cat}(\pi_i) \quad \text{where} \quad \pi_i = \{\pi_{i,1}, \cdots, \pi_{i,K}\} \]

Initial state distribution \( p(x(0)) \) is a categorical distribution
\[ x(0) \sim \text{Cat}(\pi_0) \quad \text{where} \quad \pi_0 = \{\pi_{0,1}, \cdots, \pi_{0,K}\} \]

state emission model \( p(y(t)|x(t)) \) can be any distribution with parameter set \( \phi \) depends on hidden state
\[ y(t)|x(t) = i \sim f(y(t); \phi_i) \]

parameters to be learned are \( \theta = \{\phi_i, \cdots, \phi_K, \pi_0, \pi_i, \cdots, \pi_K\} \)
Techniques for learning HMM parameters

- Baum-Welch Algorithm
  - Iterative EM algorithm for estimation of ML parameters
  - Categorical observation model
- Cappe Online EM Algorithm
  - Recursive estimation of ML parameters
  - Sufficient Statistic
  - Exponential family observation model
Learning state flow models

- Techniques to learn state flow models can be looked as interpolators of sparse observations in multidimensional functions with multidimensional domains.

- Techniques have to face dimensionality and probabilistic aspects. Examples are:
  - Methods that learn flow models (e.g., Gaussian Processes).
  - Methods that learn flow models and simultaneously estimate superstates using generative approaches.
  - Methods that learn also probability transition of superstates in addition to above steps.
Non-linear dynamics

\[ x_t = f(x_{t-1}) + \omega_t \]

\[ \omega_t \sim \mathcal{N}(0, \sigma) \]
Bayesian non-linear regression

suppose we want to predict value of an arbitrary function $f(x) : \mathcal{X} \rightarrow \mathbb{R}$

using a set of samples $D = \{x, y\}$

where $\mathcal{X} = \{x_1, \cdots, x_N\}$ and $y = [f(x_1) + n_1, \cdots, f(x_N) + n_N]^T$

$n_i \sim \mathcal{N}(0, \sigma)$ is measurement noise

Using Bayes’ rule:

$$p(f|D) = \frac{p(D|f)p(f)}{p(D)}$$
Bayesian non-linear regression

Gaussian process defines a distribution over functions $p(f)$:

let $\mathbf{f} = [f(x_1), \cdots, f(x_N)]^T$ be a vector of function values evaluated at $N$ arbitrary points, $p(f)$ is a Gaussian process if $p(f)$ has a multivariate Gaussian distribution.

Gaussian process is characterized by a covariance function (kernel) $\kappa(x, x')$

$$p(f(x_1), \cdots, f(x_N)) = \mathcal{N}(0, \Sigma_{\cdot \cdot})$$

where

$$\Sigma_{\cdot \cdot} = \begin{bmatrix}
\kappa(x_1, x_1) & \kappa(x_1, x_2) & \cdots & \kappa(x_1, x_N) \\
\kappa(x_2, x_1) & \kappa(x_2, x_2) & \cdots & \kappa(x_2, x_N) \\
\vdots & \vdots & \ddots & \vdots \\
\kappa(x_N, x_1) & \kappa(x_N, x_2) & \cdots & \kappa(x_N, x_N)
\end{bmatrix}$$
Bayesian non-linear regression

Example of covariance function:

$$\kappa(x_i, x_j) = \theta_1 e^{-\frac{(x_i - x_j)^2}{\theta_2}}$$

parameters of covariance function, \(\{\theta_1, \theta_2\}\), are estimated using ML gradient descent algorithm
Bayesian non-linear regression

• Prediction of function values

suppose we want to predict function value $f^* = [f(x_1^*), \ldots, f(x_M^*)]^T$

Given GP prior the function values are jointly gaussian

$$p(f^* | D) = \mathcal{N}(\bar{f}^*, \Sigma_{f^*})$$

$$\bar{f}^* = \Sigma_{*|}.[\Sigma_{.}. + \sigma^2 I]^{-1}f$$

$$\Sigma_{f^*} = \Sigma_{*|}* - \Sigma_{*|.}[\Sigma_{.}. + \sigma^2 I]^{-1}\Sigma_{.|*}$$

where

$$\Sigma_{*|.} = \Sigma_{T|.} = \begin{bmatrix} \kappa(x_1^*, x_1) & \cdots & \kappa(x_1^*, x_N) \\ \vdots & \ddots & \vdots \\ \kappa(x_M^*, x_1) & \cdots & \kappa(x_M^*, x_N) \end{bmatrix}, \quad \Sigma_{*|*} = \begin{bmatrix} \kappa(x_1^*, x_1^*) & \cdots & \kappa(x_1^*, x_M^*) \\ \vdots & \ddots & \vdots \\ \kappa(x_M^*, x_1^*) & \cdots & \kappa(x_M^*, x_M^*) \end{bmatrix}$$
Bayesian non-linear regression

• Conclusion

  • Gaussian Process allows to estimate values of an arbitrary function at some evaluation points given noisy samples of the function.
  • It gives estimation and uncertainty (error bars) of prediction
Bottom up incremental learning of state flow models

- HMM example is not capable to capture incremental aspects as time is not explicitly considered.
- The most CDS suited approach is to incrementally jointly learn from state observations in a bottom up way.

- \( p(x_t/x_{t-1}, s_t) \)
- \( p(s_t/s_{t-1}) \)
Goals for a ID-CDS: Learning in PGM CDS models

- How can knowledge be incrementally updated in the PGM at each step a new observation or an action is done by the CDS?
- To learn models in a unsupervised and incremental way one could choose to use directly as input observations $Z_{1:k}$ at step k.
- However, this could be really useful only when one has to jointly estimate also observation model.
- If observation model is known and assumed to be stable in time, incremental acquisition of observations $Z_{1:k}$ after a reference prediction step can make new state estimation $X_{1:k}$ available.
Goals for a ID-CDS: Learning in PGM CDS models

- So application of a reference filter can generate a sequence of state values that can be used to learn above switching models.
- This is equivalent to start from a very simple dynamic model.
- \( \text{pref}(x_t/x_{t-1}) \)

E.g. the random walk with Gaussian noise.

- The covariance error of reference filter can be used as a measurement to perform contextual incremental flow state model and vocabulary update based on a Cognitive Control like scheme.

- A stick breaking or similar Dirichlet based approach can be used to manage non stationarities and anomalies wrt previous knowledge. This happens when available state flow models do not describe current patterns.
Examples
Examples: case studies

Trajectory based incremental state flow model learning

- Non linear
- Piecewise linear
Conclusions

- PGM based -Cognitive Dynamic systems are here defined as based on the capability to represent Self awareness and Cognitive Control.
- PGM allow to represent SA and CC in a hierarchy of coupled dynamic stationary dynamic models in a coherent framework.
- Semantic and subsemantic levels are both represented.
- PGM can be incrementally learned from experience.
Special thanks to

- Vahid Bastani and Damian Campo for supporting slides
- Lucio Marcenaro for slides from other tutorials jointly done
- ISIP40 PhD students carrying on such research in last 15 years.
Thanks for your attention!

Contacts:

- Carlo Regazzoni (carlo@dibe.unige.it)
- Via Opera Pia 11, 16145 Genova (ITALY)
- Webpage: http://www.isip40.it/