i^2 Learning : Overcoming Mismatching & Asymmetric Inconsistencies

Yogesh R. Isawe

Contents

Introduction

- Primary objective of intelligent agent systems is problem solving.
- When an agent system can adequately handle its tasks, learning may not be needed.
- But when the agent encounters a conflicting circumstance for which it does not know how to handle, it needs to switch to a learning episode.
- The learning process enables agents to improve their problem-solving performance over time.
- Learning in episodic stimulus specific method is more productive to environment than that of continuous learning process.
- Inconsistencies, contradictions, conflicts or surprises can be used as stimuli to learning because they often signify; the boundaries of an agent’s know-how.

Problem Statement

- Developing stimulus-specific inconsistency learning algorithms and heuristics for the framework of perpetual learning agents for;
    1. Asymmetric inconsistency
    2. Mismatching inconsistency
- Identifying real world problem/ scenario where such inconsistencies may occur

Machine Learning

- Definition:

  A machine learns with respect to a particular task T, performance metric P, and type of experience E; if the system reliably improves its performance P at task T, following experience E. This machine learning definition has three first-class entities: (T, P, and E) in learning.

- One-time learner: LOAN (Learn Once Apply Next)
  1. Not adequate in dealing with many real world problems
  2. Its in contrast with the human's lifelong learning process

Perpetual Machine Learning

- Why Perpetual Learning ?

  Most of the agent systems do not come equipped with a complete set of problem solving knowledge.

  Learning algorithm in LOAN is used to produce a model or a target function *h* that is put to work at task T.

  No continual refinement or augmentation of the problem-solving knowledge

  Thus when the agent encounters, a conflicting circumstance for which it does not know how to handle, it needs to patch/ stretch the boundaries of its knowledge

Perpetual Machine Learning-Definition

- Learning agent relies on a set of stimuli for its successive learning episodes
- To define such perpetual learning agent system, elevate learning stimuli of first class, by adding a set of learning stimuli ‘S’, to the first-class status such that perpetual learning can be formulated in terms of (T, P, E, and S).
- Definition:

  Given T as a set of tasks, P as performance metric, E as type of experience, and S as a set of learning stimuli, a computing system perpetually learns with regard to (T, P, E, S) if the system consistently and continuously improves its performance P at T, following E and S over time.

Inconsistencies-Learning stimuli

- Inconsistencies, contradictions, conflicts, peculiarities, anomalies, outliers, or surprises often signify, the boundaries or gaps of an agent's knowledge base
- Identifying inconsistencies explicitly helps initiate the next learning episode
- Learning episode relies on the inconsistency-specific heuristics to revise, refine or augment existing knowledge to adopt to the emergent pattern and behavior
- As a result, the agent is able to patch or stretch the boundaries of its knowledge with regard to the inconsistent circumstances such that it knows how to handle the similar situation when it arises next time

Inconsistencies-Definition

- <u>Asymmetric inconsistency:</u>

Definition:

If an agent's decision or reasoning process creates asymmetric literals, then there exists asymmetric inconsistency.

Let a predicate *P* representing a symmetric relation,

We have, $\boxed{\forall x\ \forall y\ [(P(x, y) \supset P(y, x)) \wedge (P(y, x) \supset P(x, y))]}$

Let, $\boxed{L1 = P(x, y)}$ and $\boxed{L2 = P(y, x),}$

L1 and L2 are referred to as *symmetric*

If *L1 and L2* are assertions about a *symmetric relation*,

But L1 and L2 are no longer *symmetric*, we say that *L1 and L2* are *asymmetric*

For $\boxed{L1 = P(x, y)}$ and $\boxed{L2 = \neg P(y, x),}$

Asymmetric inconsistency: Example

For instance,
If L1=Connected (agent1, agent2) and
L2=Connected (agent2, agent1) then
L1 is said *symmetric to* L2

But when L1 =Connected (agent1, agent2) and
L2 = ¬Connected (agent2, agent1) then
L1 and L2 are asymmetric literals.

Inconsistencies-Definition

- Mismatching inconsistency:

  Definition:

  If an agent's decision or reasoning process gives rise to a mismatching compound predicate and its defining logical expression, then there exists mismatching inconsistency.

  Consider a compound predicate (L) is fully defined through a logical expression of other predicates (L_1).

- For instance, a mobile agent can be defined as one that can be executed on two different hosts, consider a compound predicate (L) be;

$$\boxed{\text{Mobile agent}(x) = [\text{Executing}(x, host1) \wedge \text{Executing}(x, host2) \wedge host1 \neq host2]}$$

- And logical expression that fully defines it be (L_1);

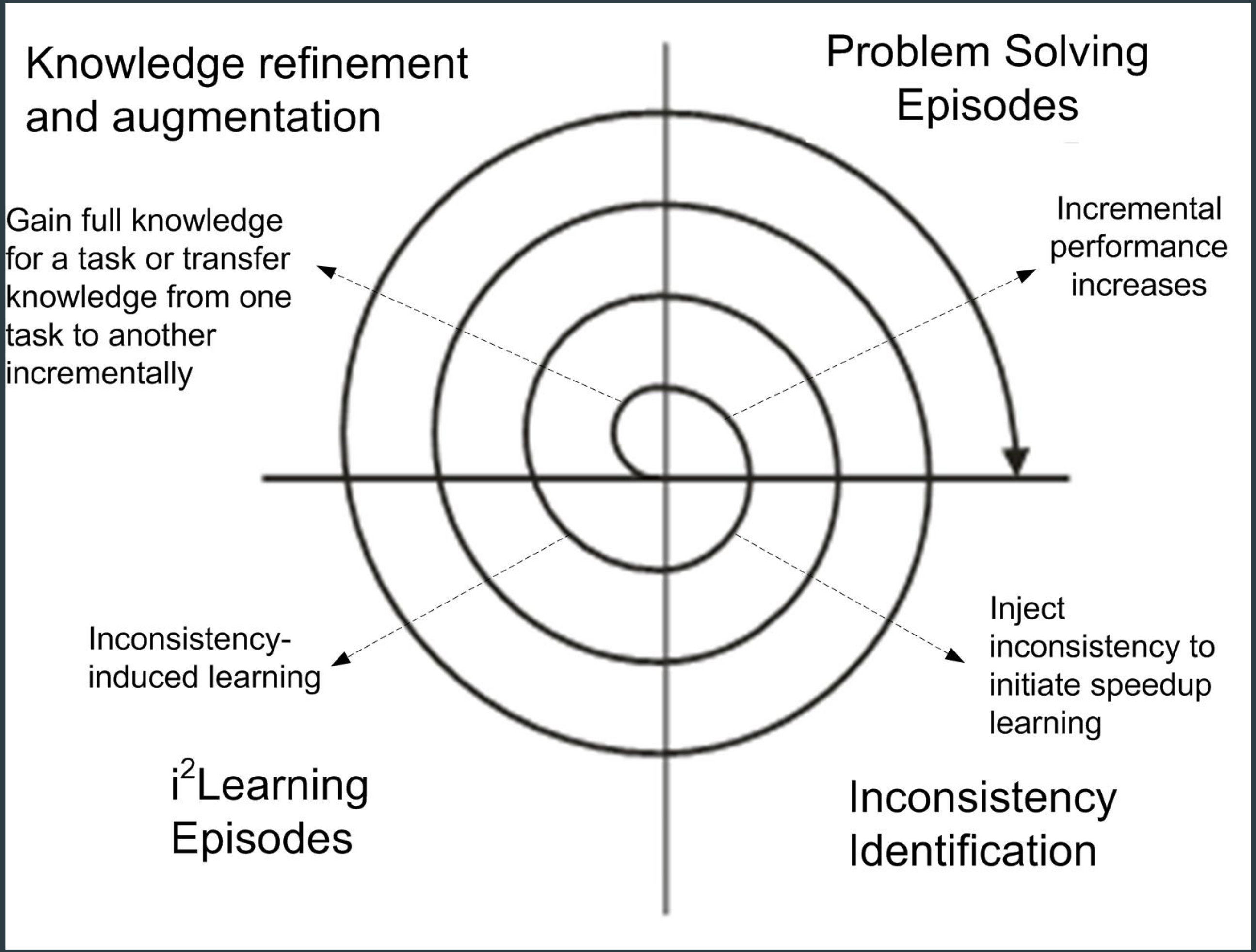
$$\boxed{(L_1) = [\text{Executing}(x, host1) \wedge \text{Executing}(x, host2) \wedge host1 \neq host2]}$$

Continued…

- Given a compound predicate (L) and

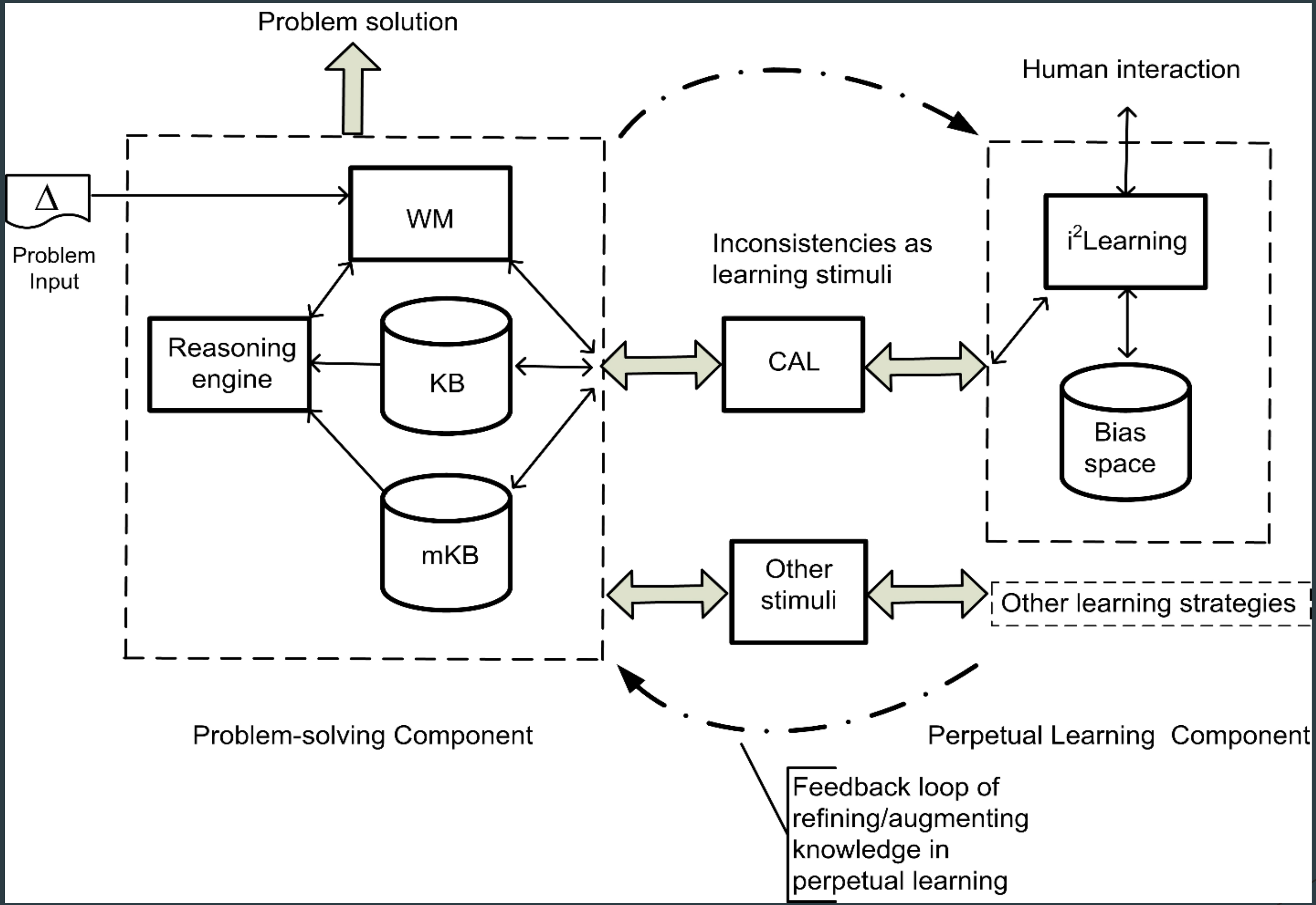
  it's fully defining logical expression (L_1),
- If we have either of the following circumstances:
  1. If $\boxed{\textit{Mobile agent } (x) \wedge \neg \textit{Executing } (x, \textit{host1})}$ *OR*
  2. If $\boxed{\neg \textit{Mobile agent}(x) \wedge [\textit{Executing}(x, \textit{host1}) \wedge \textit{Executing}(x, \textit{host2}) \wedge \textit{host1} \neq \textit{host2}]}$
- Then we say that the compound predicate and its defining logical expression are *mismatching*.

Framework-Spiral Model



Spiral model of episodic and perpetual learning

Framework-i^2 Learning



Heuristics and algorithms

- <u>Mismatching inconsistency:</u>
- Mismatching inconsistency refers to mismatched derivation of a compound predicate (L) and it's fully defining logical expression (L_1), denoted as $L \neq L_1$.
- To facilitate learning through overcoming mismatching inconsistencies, we define;
  1. $\mathfrak{I}$: set containing conflicting circumstances expressed as pairs of mismatching literals
  2. $\aleph$: set of heads (conclusions) of the rules that explain the inconsistency
  3. $\wp$: set of predicates that appear in WM (working memory)
  4. $\mathfrak{R}$: set of refined rules produced at the end of a learning

Continued…

- Let KB_0 contains knowledge;

  $Purchase(x) \longleftarrow up\_trade(x)$

  $\neg\ Purchase(x) \longleftarrow \neg up\_trade(x)$

  $up\_trade(x) \longleftarrow Good\_standing\_Share(x)$

  Good_standing_Share(s1)

  Good_standing_Share (s2)

- Now, WM_0 has following facts;

  Purchase (s1)

  ¬ Purchase (s2)

  ¬Daily_drop (s1, 15%)

- CAL detects that there exists the following mismatching inconsistency in $KB_0 \cup WM_0$

  $\boxed{KB_0 \cup WM_0 \vdash \{Purchase\ (s2),\ \neg\ Purchase\ (s2)\}}$

Continued…

- The aforementioned complementary inconsistency would cause CAL to initiate an episode of learning. The input passed to the learning module from CAL consists of the following:

  $\mathfrak{I}$ = {Purchase (s2), ¬ Purchase (s2)}

  $\wp$ = {up_trade, ¬Daily_drop}

  $\aleph$ = {Purchase}

- The learning will refine the “Purchase” rule into the following:

  $\boxed{\mathfrak{R} = \{Purchase\ (x) \leftarrow up\_trade\ (x) \wedge \neg Daily\_drop\ (x, 15\%)\}}$

- Thus KB_1 now contains the following:

  Purchase (x) ← up_trade (x) ∧ ¬Daily_drop (x, 15%)

  ¬ Purchase(x) ←¬up_trade(x)

  up_trade(x) ←Good_standing_Share(x)

  Good_standing_Share(s1)

  Good_standing_Share (s2)

Heuristics and algorithms

- <u>Asymmetric inconsistency:</u>

  Let *L1* = *Purchase (share1, share2)*

  and *L2* = *Purchase (Share2, share1)*,

  L1 and L2 are referred to as *symmetric*, If *L1 and L2* are assertions about a *symmetric relation*.

- But if *L1 and L2* are no longer *symmetric*, we say that *L1 and L2* are *asymmetric*.

  For instance, when *L1* = *Purchase (share1, share2)*

  and *L2* = *Purchase ¬ (Share2, share1)*,

  then L1 and L2 are asymmetric literals.

Continued…

- To facilitate learning through overcoming asymmetric inconsistencies, we define;
- Two literals *P* and *Q*, lets say having *precedence* relationship with some supporting function say η.
- We use ϖ (*P*) and ϖ (*Q*) to denote the *weight* information (expressed in terms of priority, importance, or significance, etc.) for *P* and *Q*, respectively.
- And we use φ (*P*) (or φ (*Q*)) to denote the fact that the presence of *P* (or *Q*) violates some domain constraint.
- Given an instance of asymmetric inconsistency;

  $\mathfrak{I} = \{P(a, b), \neg Q(b, a)\} \vee \{\neg P(a, b), Q(b, a)\}$ where *P* and *Q* are asymmetric,
- $\mathbb{C} = \{P(x, y), Q(y, x)\}$ (or $\mathbb{C} = \{P(x, y), \neg Q(y, x)\}$ or $\mathbb{C} = \{\neg P(x, y), Q(y, x)\}$)

Continued…

- We use the following notation to denote that the support function above the line is *preferred* over the support function below it with regard to the circumstance as specified by $\mathbb{C}$:
- *P and Q* are symmetric : $\{P,\ Q\}$ $\quad \frac{\mathbb{C}:\ \eta\ (P)}{\eta\ (Q)}\ ,\quad \frac{\mathbb{C}:\ \eta\ (Q)}{\eta\ (P)}$
- *P and Q* are asymmetric : $\{P,\ \neg Q\}$ $\quad \frac{\mathbb{C}:\ \eta\ (P)}{\eta\ (\neg Q)}\ ,\quad \frac{\mathbb{C}:\ \eta\ (\neg Q)}{\eta\ (P)}$

  $\{\neg P,\ Q\}$ $\quad \frac{\mathbb{C}:\ \eta\ (\neg P)}{\eta\ (Q)}\ ,\quad \frac{\mathbb{C}:\ \eta\ (Q)}{\eta\ (\neg P)}$

- Thus, given the $\mathfrak{I}$ (interpretation) containing asymmetric inconsistency. Circumstances to choose support function over another depends on;
  1. Subclass – superclass generality
  2. Most recent function
  3. Higher priority
  4. Violation of domain constraint
  5. Derivability from KB

Conclusion

- Defined machine learning and perpetual learning with first class entities
- Introduced basic concepts regarding perpetual learning agent system
- Role of inconsistencies in perpetual learning
- We described framework for developing perpetual i^2 Learning agent system
- Discussed inconsistency-induced learning strategies that refine or augment knowledge so as to improve agents performance

References

- [1] T. Mitchell, Machine Learning, McGraw-Hill, New York, NY, 1997
- [2] T. Mitchell, The discipline of machine learning, Technical Report
- [3] D. Zhang, Perpetual learning through Overcoming Inconsistencies, ICTAI 2013
- [4] D. Zhang, Learning through Overcoming Inconsistencies
- [5] Du Zhang and M. Lu, Inconsistency-induced learning for perpetual learners, *International Journal of Software Science and Computational Intelligence*, 2011.
- [6] D. Zhang and M. Lu, Learning through overcoming inheritance inconsistencies, *Proc. of 13th IEEE International Conference on Information Reuse and Integration*, Las Vegas, NV, August 2012.
- [7] D. Zhang, Learning through overcoming incompatible and antisubsumption inconsistencies, *Proc. of the 12th IEEE, International Conference on Cognitive Informatics*, New York City, NY, July 2013.
- [8] D. Zhang, Perpetual learning through Overcoming Inconsistencies, ICTAI 2013