Belief Revision Assignment

April 2024

Introduction to AI
02180

Irene Valentina Berganzo s223230
Jiwoo Eom s231567
Sara Riegels Trangbæk s174932
Tobias Sætervoll Vangsy s234113

Contents

1 Introduction

1.1 Overview of belief revision

Belief revision is a crucial component of artificial intelligence, involving the dynamic modification of beliefs held by an intelligent agent when new information is presented. Belief revision aims to sustain a consistent and accurate repository of knowledge, used for decision-making processes. The agent should be able to navigate constantly changing environments, where the new beliefs possibly contract with existing beliefs.

The belief revision agent works as following. It contains a set of logical beliefs with information. When new beliefs are passed, the agent revises the information base to remove conflicting beliefs and add the new information. This is called contraction and expansion. The process is designed to maintain the coherence of the belief base amid new information or changes in the environment, making sure that the system is both relevant and reliable.

2 The belief revision agent

2.1 Design and implementation of belief base

In developing our belief base for the AI project, we employed Python as our primary programming language due to its flexibility and robust support for logical operations. Our implementation revolves around the manipulation of tuples representing beliefs within the belief base list.

Beliefs within the belief base are represented as tuples containing two elements: the propositional expression and its associated plausibility value. This representation allows for the seamless integration of beliefs into the belief base and facilitates efficient reasoning and inference processes.

The belief base is structured as a Python list, with each element comprising a tuple representing a belief along with its plausibility value. Throughout the execution of the program, this list dynamically evolves to accommodate changes in the agent's beliefs. Additionally, we employ a queue mechanism to manage pending changes to the belief base, ensuring systematic processing and maintaining the integrity of the belief base.

A key aspect of our implementation involves detecting and resolving contradictions within the belief base. We systematically compare new beliefs with existing ones to identify direct contradictions. In cases where a contradiction is detected, we prioritize beliefs based on their plausibility values, ensuring the retention of the most credible beliefs while resolving conflicts appropriately.

Our implementation also addresses implications within the belief base, allowing for the representation of logical entailments. When a new belief involves an implication, we systematically resolve it and check for consistency with existing beliefs. This process ensures the seamless integration of implications into the belief base while preserving logical coherence.

2.2 Checking logical entailment

When working with belief bases, it is crucial to examine if the new beliefs added are entailing with the ones already existing. Logical entailment makes sentences follow other sentences logically, and it is defined as $KB \models \alpha$; the sentence KB entails α if and only if in every model in which KB is true, α is also true: $KB \models \alpha$ if and only if $M(KB) \subseteq M(\alpha)$ [1]. The most intuitive way to proceed in the analysis is using truth-tables for propositional logic. However, it is a very inefficient method when it comes to large set of formulas. Therefore, the resolution algorithm has been implemented (to show that $KB \models \alpha$, we show that $KB \wedge \neg\alpha$ is unsatisfiable). The inputs requiered are the knowledge base KB and the query α sentence that wants to be checked. The algorithm follows the next steps:

1. Convert $KB \wedge \neg\alpha$ into conjunctive normal form (CNF). It will ensure they are represented as sets of clauses, being disjunction of literals.

2. Iterate between pair of clauses to generate new clauses applying resolution rule; if two clauses contain complementary literals (a literal and its negation), it is resolved to produce a new clause, and added to the set if it is not already there.

3. Terminate whenever a empty clause is derived, or when no new clauses are generated.

This method is suitable since it always terminates, and it is also complete by the ground resolution theorem (stating that if a set of clauses is unsatisfiable, then the resolution closure of those clauses contains the empty clause) [1].

2.3 Contraction of belief base

Contraction is the act of removing the beliefs from the belief sets. This can be necessary either because new information in the belief sets contradict each other or simply because new information is given to the system (Information that results in "x" not being possible/-believed). In this project the contraction is based on the priority that the user specifies for each belief, which is then ordered every time new information is added to the belief sets. The information with the highest priority are should be kept, while the ones with lower priority are removed from the belief sets. Thus, contraction is basically the act of removing information from the belief base.

The way contraction works is as follows:

1. First the belief base is defined/initialized. The belief base will be made up of logic and propositions expressed mathematically.

2. Then the expressions/information to be removed is specified
3. Next step is to check how the information that wants to be removed is connected with the other information in the belief sets
4. Based on the priority given to each information, information with higher priority should be kept, if possible
5. The specified entity is removed from the belief base
6. The step above might create logical flaws that need to be handled accordingly
7. Final step is to check that the updated belief base is consistent

2.4 Expansion of belief base

Expansion of the belief base refers to the process of adding new beliefs or modifying existing beliefs in the belief base. The expansion is a managed process that ensures the belief base remains consistent, relevant and useful for the tasks it supports. In this project the new logical beliefs are held against all other beliefs in the belief base, after checking for entailment. If the new belief is not contradicting any of the other beliefs in the base, it is added. If it contradicts a belief in the base the one with the highest plausibility is then added and the other removed.

2.5 Output

The wishful output is a consistent and reliable belief base containing only relevant beliefs. The reality of the belief revision engine created for this project is a different story. The logical resolution is flawed which is tested. For future work, we need to revise these points and the system will be completed in the direction mentioned above.

3 What have we learned

Throughout this project, we gained valuable insights into belief revision and its integration into artificial intelligence frameworks.

One significant aspect was the fusion of Propositional Logic with Formal Methods. This project provided a practical opportunity to blend concepts from propositional logic with formal methods like resolution, CNF-form, AGM revision, and partial meet contraction. This synthesis was critical for crafting a robust belief revision engine.

We deepened our understanding of resolution-based methods for logical entailment. Resolution, combined with proof by contradiction, emerged as a crucial tool for establishing logical consequences between sentences and maintaining consistency within the belief base. Exploring techniques for handling implications within the belief base underscored the importance

of resolving implications and ensuring consistency to avoid conflicts and preserve coherence in the agent's knowledge representation.

The utilization of AGM postulates, including the Success postulate, Inclusion postulate, Vacuity postulate, Consistency, and Extensionality, offered a theoretical framework for evaluating the algorithm's effectiveness. These postulates guided both the design and assessment of the belief revision engine, ensuring alignment with rationality principles.

We placed particular emphasis on maintaining mathematical precision and technical rigor throughout the implementation and report-writing process. By employing precise language and formal concepts from the course, we enhanced the clarity and accuracy of the project's documentation. Overall, this project offered invaluable hands-on experience in applying theoretical concepts of belief revision to real-world scenarios, emphasizing the importance of logical consistency and rationality in AI systems.

4 Conclusion and further work

Throughout this project, we have endeavored to construct a comprehensive belief revision engine capable of managing and updating an agent's beliefs in response to changing information. By implementing key functionalities such as belief prioritization, logical entailment checking, and mechanisms for contraction and expansion, we have established a solid foundation for the development of intelligent systems that can reason and adapt effectively.

Our belief base architecture, coupled with resolution-based entailment checking, provides a robust framework for evaluating the consistency and coherence of beliefs. The integration of prioritization mechanisms allows for the management of conflicting information, ensuring that the most relevant and credible beliefs are retained and updated accordingly. Furthermore, our approach to handling implications enhances the expressiveness of the belief base, enabling the representation of complex logical relationships.

While the current implementation demonstrates notable achievements, there remain avenues for further exploration and refinement. One such area is semantic transformation, where the ability to seamlessly translate between informal and logical representations would enhance the versatility and interoperability of the belief revision engine. Additionally, the extension of logic to incorporate existential, universal, and nested quantifiers would elevate the sophistication of reasoning capabilities, enabling more nuanced analysis of complex scenarios.

Performance optimization is another critical consideration, particularly as the scale and complexity of the belief base increase. Efforts to enhance efficiency and scalability will be essential to ensure that the belief revision engine remains responsive and viable in real-world applications, even when confronted with extensive datasets and computational demands.

Furthermore, user interface enhancements represent an opportunity to improve the accessi-

bility and usability of the belief revision engine. Intuitive visualization and interaction tools can empower users to navigate and manipulate the belief base with ease, fostering greater engagement and understanding of the underlying reasoning processes.

In conclusion, while this project marks a significant milestone in the development of a belief revision engine, it also represents a springboard for future innovation and advancement. By addressing the identified areas for improvement and continuing to explore emerging technologies and methodologies, we can unlock the full potential of intelligent systems capable of adaptive reasoning and decision-making in dynamic environments.

References

[1] Russell, Stuart and Norvig, Peter. *Artificial Intelligence A Modern Approach, third edition.* 2010.