A Framework of Computational Opinions

A Thesis
Submitted to the Faculty
in partial fulfillment of the requirements for the
degree of

Doctor of Philosophy

by

Fei Yu

Thayer School of Engineering
Dartmouth College
Hanover, New Hampshire

May 2013

Examining Committee:

Chairman ____________________
Professor Eugene Santos Jr.

Member ____________________
Professor George Cybenko

Member ____________________
Professor Mark Borsuk

Member ____________________
Professor Joseph M. Rosen

Member ____________________
Professor Hien Nguyen

____
F. Jon Kull
Dean of Graduate Studies

ABSTRACT

We have a limited understanding of how an opinion is originated, how an opinion and information supporting and explaining it gets conveyed, and how the communicated opinion is perceived and processed by others. One direction of current research focuses on the conditions and determinants for opinion formation, while another focuses on opinion change induced by external influence (social influence in particular), however, empirical findings have concluded that the persistence of opinion change is inconsistent. This indicates that current methods used to predict the next opinion based on the the current opinion may fail if the decay in opinion change is rapid.

We realize that for the existing computational framework/models, the close interplay between opinion formation and change is not exploited well. Prevalent computational frameworks/methods either model only one opinion formation process or only one opinion change process. Furthermore, computational frameworks/models that actually focus on opinion dynamics either only model one task or model a sequence of tasks but do not differentiate between them.

Our insight to better address this problem is that we recognize the learning nature of opinion change and the decision-making nature of opinion formation: what has been learned through internalizing an external influence guides how decisions are made to externalizing cognitive processes. Thus the challenges for building a computational framework lies in modeling both the learning and decision-making aspects for the entire opinion formation task.

To address this modeling challenge, we also recognize that the individual performing an opinion formation task is essentially engaged in a sequential decision-making problem with a specific goal in mind. This thesis presents our effort in two phases: In Phase I, we propose a networked space of reasoning processes which we call the Double Transition Model (DTM). Each node within a DTM represents a cognitive state based on different degrees of query and knowledge incompleteness. The edges within

a DTM denotes how query and knowledge differ between the connecting states which can be fulfilled to trigger changes in an opinion. To ensure that diversification in opinion formation processes can be covered by a DTM, we evaluated it by simulating four commonly accepted heuristics in human reasoning.

The edges in a DTM are designed for modeling influence. In Phase I, we do not differentiate the causes of the influence - whether it is internally embraced or externally activated, so a DTM is mainly a cognitive model. In Phase II, we extend a DTM into a knowledge-based behavior model. How individuals perceive messages, accept messages, and communicate messages is simply a sequence of decisions. Therefore, we can construct a computational framework of opinions in which an opinion formation task can be formulated as a Markov decision process defined using an augmented DTM. Furthermore, we are able to use methods such as Q-learning to free us from the requirements of having full knowledge of communication dynamics in deriving optimal policies. In a case study, we demonstrate the power of the framework in making the entire opinion dynamics between two individuals analyzable.

Acknowledgements

First of all, I would like to thank my advisor Professor Eugene Santos Jr. He has been an ideal image of a researcher and guided me into the wonderland of research. By working with him I learned the value of vision, the importance of critical and independent thinking, and the value of persistence. He has never suggested to me what type of researcher I should be or I can be - being empirical, theoretical, or conceptual - instead he let me discover myself through my own thesis work. Without his tremendous patience and guidance this thesis would have not been realized.

Thanks also to my committee members: Professor George Cybenko, Professor Mark Borsuk, Professor Joseph M. Rosen, and Professor Hien M. Nguyen. Professor Cybenko planted a seed about Markov decision processes in my brain through his classes that made this framework of computational opinions possible. Professor Borsuk taught me about the diversity, uncertainty, and complexity of human behavior which motivated me to work on this framework. Professor Rosen showcased many real medical cases that had strengthened my beliefs about the value of a computational framework and indeed the needs to understand/model human behavior/intent. Professor Nguyen has taught me so many research skills from writing a paper to conducting empirical research from the general picture to every small detail that could possibly improve the overall quality of research.

I also would like to thank my fellow students. Dr. Keum J. Kim taught me the importance of stating assumptions and why details matter. Her skeptical eyes towards research constantly remind me to be as skeptical and cautious as possible especially towards my own work. Tracy taught me the beauty of math and its power in defining a framework in a more rigorous and accessible way. Thanks to Jacob, who shared with me a lot of the advances in programming which has substantially increased my productivity. Thanks to Jeremy, Curtis, Jake, Yan, and many many previous fellow students. We all shared a great journey along with each other. Every one of you is a genius in a certain way and each of you have inspired me to look at things from a different angle. I would never forget about the numerous heated discussions we had in the group meetings and the long-hours we had together for the group projects. Lastly but importantly, thanks to Yuki with whom we shared our research journey all along for these six years. It is amazing that at the end we turned out to be really different types of researchers but yet we share extremely similar understanding towards research.

Thanks everyone who has tolerated me when I seemed very empty or brain-occupied. Thanks for the support from Louise, Thayer School of Engineering, and all the research grants that made this happen.

Finally, I would like to thank my parents who have never asked me when I would graduate. All they have said repeatedly to me is: "Do it. Try all your best, but take good care of youself."

To my parents.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Opinions are *personal beliefs* (Krueger, 1996) - an opinion is belief as it is derived rather than recalled; an opinion is personal as it can be derived in diverse ways. Studying opinions is particularly appealing to both practitioners and researchers because: 1) one's opinions can complement others' knowledge (Pang and Lee, 2008), 2) one's knowledge can complement others' experiences (Danescu-Niculescu-Mizil et al., 2009), and, 3) one's opinions can complement and enhance other's thinking (Bindel et al., 2011; Clayton, 1997).

Despite the great interest the research community has in opinion mining (particularly in sentiment analysis (Archak et al., 2007; Bethard et al., 2004; Breck et al., 2007; Pang and Lee, 2008)), we still have a limited understanding of how an opinion is originated, how an opinion and information supporting and explaining it gets conveyed, and how the communicated opinion is perceived and processed by others. One direction of current research focuses on the conditions and determinants for opinion formation (Afshar and Asadpour, 2010; Berelson et al., 1986; Watts and Dodds, 2007), while another prevalent direction focuses on opinion change induced by external influence (social influence in particular) (Aronson et al., 1963;

Crano, 1977; Kelman, 1961). Empirical findings (Kelman, 1974; Rogalin et al., 2007) have concluded that the persistence of opinion change is inconsistent, methods of predicting the next opinion based on the current opinion may fail if the decay in opinion change is rapid (Kelman, 1955, 1961; Watts and Holt, 1979).

Our key *insight* in tackling this challenge is that we recognize the *learning nature of opinion change* (Kelman, 1961) and the *decision-making nature of opinion formation* (Clayton, 1997): the internalization of an external influence guides the externalization of internal cognitive processes. We define an opinion change process as the process of internalizing one's external influence, and we define an opinion formation process to be the process of externalizing one's internal cognitive processes[1]. We define an opinion formation task as undergoing a series of opinion formation and opinion change processes on one issue in a single context. Intuitively, an opinion formation task can include situations such as a librarian wanting to suggest a good textbook on Artificial Intelligence, congress members trying to reach consensus in a debate, or two family members deciding on the Christmas gifts to buy through an online messenger.

A framework that processes a sequence of opinion formation tasks allows us to achieve a better understanding of how external influence gets internalized (opinion change), how internal opinion gets externalized (opinion formation), and the interaction between these two processes in both short-term and long-term resolutions (short-term as in one task, long-term is a sequence of tasks). To begin with, our first objective is to model an opinion formation task. As the persistence of opinion change within an opinion formation task is stronger than the task-to-task opinion persistence, we can exploit the close interplay between opinion formation

[1]Note that the concept of an opinion formation process is a different concept from the concept of opinion formation. Both opinion formation and opinion change are more broad and general concepts.

and change processes within a task in order to learn the mechanisms of internalization.

The challenges of modeling an opinion formation task lies in modeling both the learning and decision-making aspects within an interactive environment. To address this modeling challenge we recognize that the individual performing an opinion formation task is essentially engaged in a sequential decision-making problem with a specific goal in mind.

Let us discuss what has been internalized by an opinion change process and what can be externalized by an opinion formation process. It is fairly easy to hypothesize what the external influents[2] are by considering the behavior of an individual engaged in an opinion formation task. These external influents can come in a wide variety of forms such as receiving/observing an opinion from someone, perceiving/sensing the environment for evidence, actively seeking information, the question itself that asks for opinions, an action performed, or behavior (e.g. threats (Carver, 1977), punishment and external surveillance (Hoekstra, 1995)), that can be observed from others. To simplify the problem, we concentrate on influents in the form of messages since what has been internalized is much less clear as it all happens within a human brain which, of course, lacks "visibility". A variety of social theories have identified that the internalization may include the knowledge basis (McGuire, 1968) from which an opinion is formed, a value system (Kelman, 1961), and sentiment towards others (Robinson et al., 2006). To be consistent with our simplification on the external influents, we concentrate on the underlying knowledge base and the reasoning process from which an opinion can be derived.

[2]External influents refers to messages from the world.

The overall objective of this thesis is to develop a knowledge-centric computational framework that processes sequential opinion formation tasks. We present the effort in two phases. Phase I focuses on addressing the learning aspect of an opinion formation task. We propose a networked space of reasoning processes which we call the Double Transition Model (DTM). Each node within a DTM represents a cognitive state with different degrees of query and knowledge incompleteness. The output of a reasoning process is an emitting probability representing a possible opinion. The edges within a DTM denotes how a query and knowledge differ between the connecting states. This design accounts for changes that can occur during the opinion formation process.

The edges in a DTM are designed for modeling influence. In Phase I, we do not differentiate the causes of the influence - whether it is internally embraced or externally activated so a DTM is mainly a cognitive model. In Phase II, we extended DTM to address the decision-making aspect of an opinion formation task. The behaviors specified in the extended DTM are mainly to capture the specifics of influents such as the direction of an influent (internal versus external), its content, and its action type (accept new knowledge, correct unsubstantial knowledge, or discard irrelevant information).

As we mentioned above, the individual performing an opinion formation task is essentially engaged in a sequential decision-making problem with a specific goal in mind. From this, we recognized that completing the design of our computational framework can be achieved through reinforcement learning problems that are directly defined by DTMs. We start by using Markov decision processes to compute optimal actions to take at each step based on the assumption of knowing another's DTM. Alternatively, we use Q-learning methods to solve tasks where the assumption of perfect knowledge and a static environment does not hold. Furtheremore, Q-learning methods also allow us to derive a reasonable approximation of optimal policies for opinion formation tasks engaging with a series of different individuals (one at a time) and also allows for on-line learning.

Evaluation of this framework is a challenging task due to the scope of the tasks and the lack of ground truths in the internalization process. Alternatively, we evaluated the framwork from the perspective of its capabilities in computing opinions. Additionally, we evaluated whether it succeeds in internalizing an opinion change process and whether it in turn impacts the following opinion formation process.

The contribution of this thesis is as follows:

- We provided a general and compact framework that simultaneously models two distinct processes (opinion formation and opinion change) in one general opinion formation task.
- We provided, implemented, and tested different algorithms, techniques and properties that can now be used for opinion modeling.
- The framework was constructed from external influents that are easily trackable and accessible, thus requiring less knowledge engineering effort to instantiate the framework to modeling different opinion formation processes.
- The design of the framework is compatible with theories in bounded rationality.

1.1 A Framework of Computational Opinions

1.1.1 Motivation

In the past few years, we developed computational models for capturing individual differences/preferences in topics from information retrieval (Yu and Santos Jr., 2012), in information-seeking behavior (Santos et al., 2010), and in adversary intent (Santos et al., 2008, 2012). In this thesis, we are motivated to design a framework of computational opinions after identifying a critical gap between the findings from well-established social theories and the existing computational models. Kelman (Kelman, 1955, 1961, 1974) determined that "opinion changes induced by social influence can be temporary and superficial and, by contrast, those under which such changes are lasting and integrated into the people's beliefs and value systems". He has identified the various diverse effects of social influence to opinion change including intensity, duration changes to internal beliefs, and value systems. This has triggered us to reconsider the following problem.

Problem of Poor Predictivity

Imagine an opinion sequence of an individual A

$$[lo_1], [o_2], [o_3, o_4, o_5], [o_6], \cdots, [o_n]$$

where each $o_i, i \in \mathbb{N}^+$ is a degree of belief represented by a numeric value in $[0, 1]$ observed over time. The sequence in brackets is a sequence of opinions one has formed within the same context on one issue (e.g. one has been reading a collection of new articles on a presidential candidate, or one is having a heated discussion on who should be the next president.). Each bracketed sequence is an opinion formation task. A task consisting of one opinion is a non-episodic opinion formation task, and a task consisting of multiple opinions is an episodic task.

The findings from well-established social theories have concluded that the assumption $o_5 = o_6$ rarely holds. In other words, the final opinion formed within an interactive environment may not be the same for the opinion elicitepd next. The non-persistent property of opinion change leads to the problem of poor predictivity: The final opinion concluded in an opinion formation task poorly predicts the next opinion.

Unfortunately, the assumption of *persistence* of opinion change has been implicitly made in existing computational models (Hegselmann and Krause, 2002; Weisbuch et al., 2003; Yildiz et al., 2011). One typical treatment is to learn the stability of opinion change from observations of opinion dynamics (either explicitly specified or implicitly learned by structure or parameters), but the issue is that the stability of opinion change is domain-specific and consequently, the learned model can no longer be applied to solve another task due to the varying stability across different domains.

Many computational models (Hegselmann and Krause, 2002; Yildiz et al., 2011) only consider modeling opinion dynamics within a single interchange. These approaches are not adequate for studying opinion dynamics spanning a longer time period.

To summarize, the fundamental issue with existing computational models is that the need to differentiate opinion formation tasks but modeling them in one framework/model has been overlooked. One problematic situation is a model that treats a sequence of opinion exchange activities as one formation task - despite the fact that the consensus being reached over a long span of time likely suffers from the problem of poor predictivity (Dietrich and List, 2008).

Another issue with the existing computational models is that the opinion formation process and opinion change process within an opinion formation task is treated separately/independently. In essence, there is nothing learned from an opinion change process besides the opinion value itself and there are no decisions being modeled for an opinion formation process besides the opinion value itself (Martin et al., 2005). The internalization of external influence is mainly the opinion itself which has limited coverage of what has been reported in empirical findings. Similarly, the externalization of internal cognitive processes is also again just the opinion itself.

Studying opinion dynamics over a sequence of time steps spanning multiple (non-)interactive episodes is important. We need to break apart, decompose, and study opinion dynamics over a sequence of time steps into opinion formation tasks where each task is concerned with one issue within a reasonable window of time in the same context[3]. To make it clear, two discussions each on the same issue/topic with a different individual at a different time are considered to be two separate tasks.

[3]The context we focus on is the source of external influents such as individuals or the environment.

We assume that there is a relatively high persistence in opinion change within one task. By defining a task this way, it allows us to exploit the close interplay between opinion change and formation processes within a task to learn the mechanisms of internalization. By modeling the interplay between opinion formation and opinion change processes in one task, we are then able to address the issue of poor predictivity of opinion changes resulting from one task to another task.

To recap, we are motivated to develop a computational framework of opinions to facilitate studies in opinion modeling. In particular,

1. model the decision-making nature of opinion formation and the learning nature of opinion change (within one task) (Marked by a hexagon with number 1 in Figure 1.1)

2. provide a foundation to study the persistence of opinion change in long-term projections (cross tasks) (Marked by a hexagon with number 2 in Figure 1.1)

This effort is important as the existing computational models are inadequate:

- opinion formation process and opinion change process are modeled separately and independently (corresponds to item #1 in the list of motivations above)

- a sequence of opinion formations tasks are either modeled separately or modeled together but are not differentiated (corresponds to item #2)

This thesis is developed with the core goal of modelling a sequence of opinion formation tasks. Figure 1.1 illustrates our motivation and objectives of this thesis. In the figure, each bracket (#1 in Figure 1.1) represents a sequence of activities occurring within one opinion formation task. An incoming arrow from the environment denotes external influences which may result in opinion changes. An outgoing arrow from the agent itself denotes the communication of an opinion derive from an opinion (along with other messages) formation process. The arrow from one task to another (#2 in the Figure) denotes the impacts of internalization of external influences from one task to another. The interplay between opinion change and formation in both short-term and long-term temporal resolutions are the essence of this framework. The following two sections describe the framework development of the thesis. Detailed justfications of the design of the framework will be supplied in the motivation section in the following chapters.

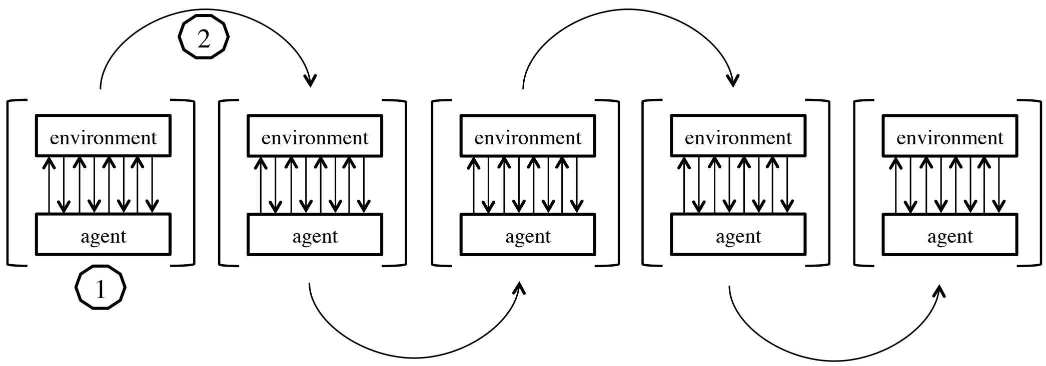


Figure 1.1: Conceptual Diagram of the Framework

1.1.2 Phase I: Double Transition Model

In Phase I (covered in Chapters 4 and 5), we developed a Double Transition Model (DTM) which is a cognitive model. A DTM is a graph derived from two independent graphs: one is a memory transition graph and the other one is a query transition graph.

Memory Transition Graph

As shown in Figure 1.2, each node in a memory graph represents a snapshot of the working memory. The entire graph stores all past histories of memory linked by perceived information. The graph starts empty and creates a node every time new information is perceived. (This approach assumes the perception of one piece of

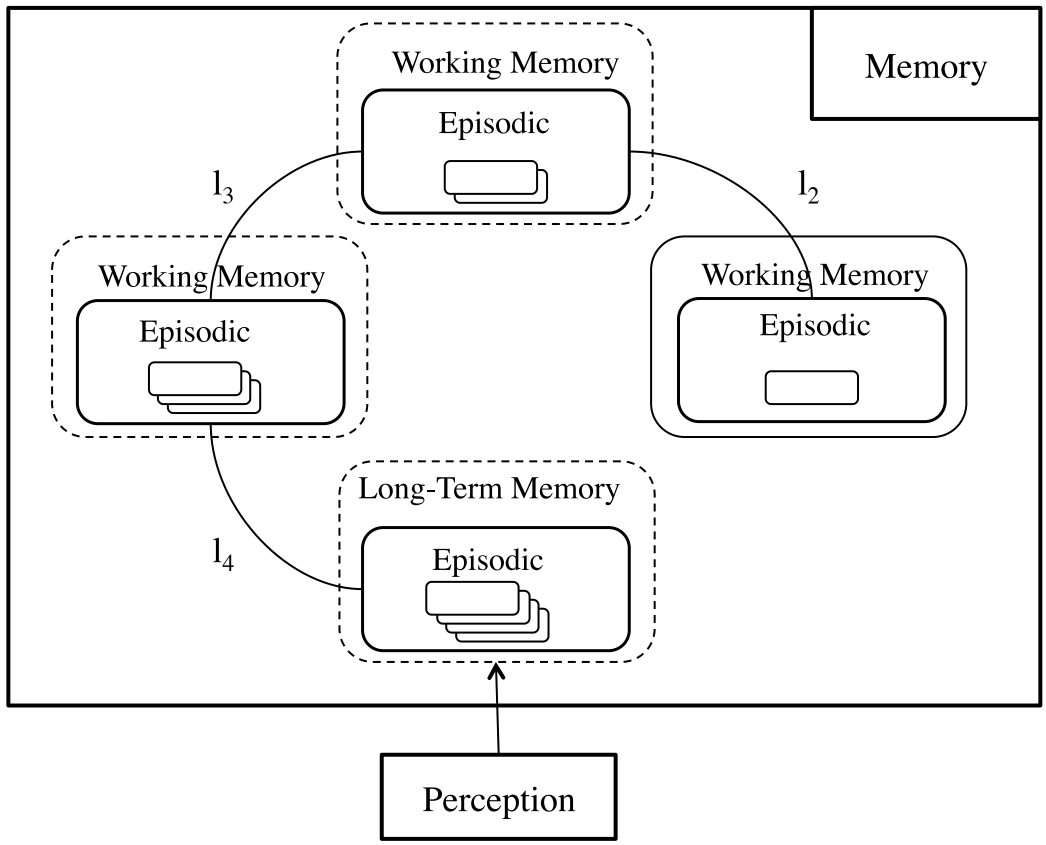


Figure 1.2: Memory Transition Graph

information at a time.) The new node will then have all the information perceived in the past plus the new information that has just been perceived. This new node will be linked to the current node. As such the difference between two connecting memories is always one piece of information that we refer to it as a learning episode.

We always keep a marker that points to the most recent memory (node) which at the same time stores the entire history of learning episodes. We call this long-term memory as it contains the most complete set of knowledge, and all the other nodes are considered to be candidates for short-term memory as it stores partial knowledge from the long-term memory.

This is a simple memory transition graph to start with. One thing we would like to emphasize is that the graph itself is not intended to indicate the way memory is organized. The purpose of this memory transition graph is to model how a memory can be formed and the edges that can be considered as stimulus for memory recall. We intend to be brief here in terms of why the memory transition graph is designed this way and how it is different from the existing memory structure. Full details and justifications for such design can be found in Chapter 4.3.

Query Transition Graph

We applied the same idea as above to the development of a query transition graph. As shown in Figure 1.3, each node in a query graph represents a query at a time. Each query is recorded when an opinion is being requested by someone else. Therefore, the creation of a new query is triggered by a task while the new creation of a memory is triggered by perception (in Figure 1.2). The entire query graph stores all past histories of queries. The graph starts empty and creates a new node for every inquiry for opinions. We also kept a marker that points to the most recent query being asked but the query is no longer accumulated as how memory is, therefore, an edge between two nodes represents the difference in the content of the connecting two queries.

The representation of a query is simply a vector such as $[x, ?, ?, 3, \cdots, 2]$ where a numeric value represent either a category or a discretized number for a feature of an entity, a question mark represents unknown value for a feature, and x represents the target feature of an entity. For example, one may ask another one for opinions about a university: "This school has tuition around $\$42,000$, yearly enrollment around $11,000$ and an acceptance rate around 7.7%, do you think this is a tier 1 school?" The other features that are not included in an inquiry are treated as unknown, and the targeted feature is thus the rank of the school. In Figure 1.3, unknown features are represented by white rectangles, known features are grey rectangles, and targeted features are represented by black rectangles.

The easiest way to understand this graph is that it is simply a sequence of opinion inquiry tasks. The edges denote the ways relevant tasks can be recalled - therefore, the form of links introduced here only consider temporal relationships as a clue to retrieve relevant tasks. We will formally present the general definition of this transition graph in Chapter 4.4.

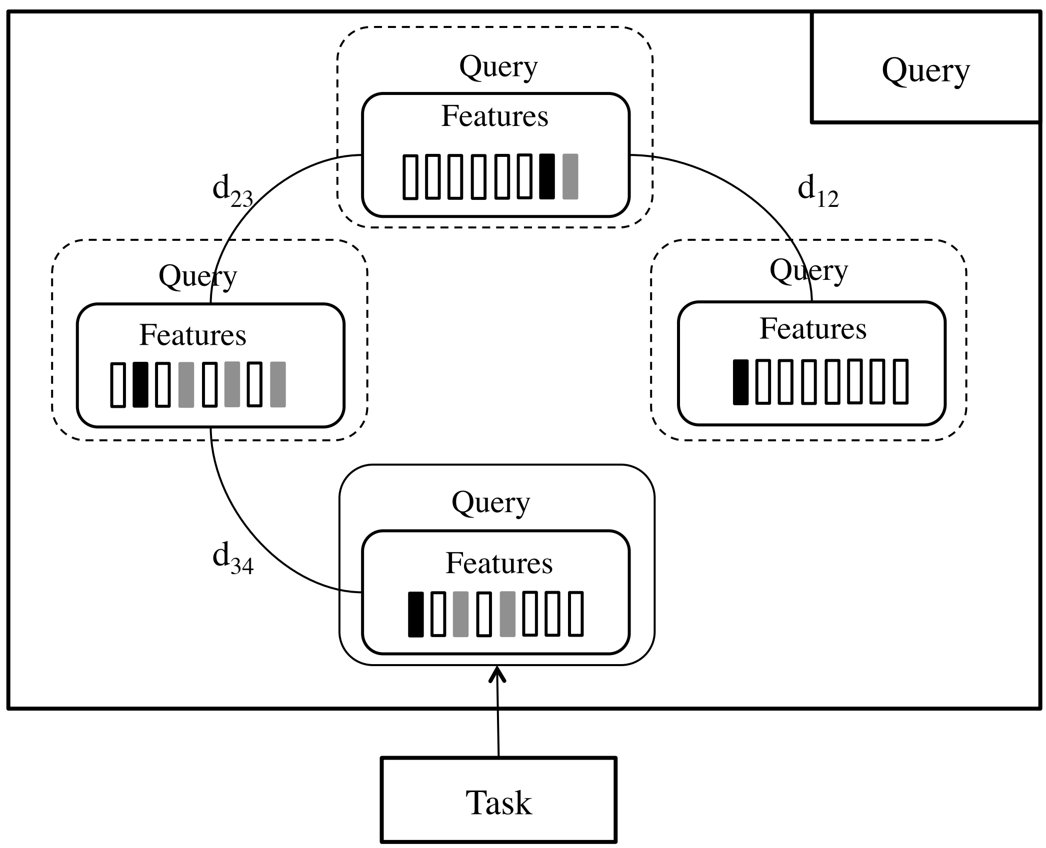


Figure 1.3: Query Transition Graph

Double Transition Model

A Double Transition Model can be derived by combining a memory transition graph and a query transition graph. As shown in Figure 1.4, each node in a DTM is a combination of a node in a memory transition graph and a node in a query transition graph. A node in a memory transition graph is a working memory (long-term memory is also a working memory with full knowledge), a node in a query transition graph is a query from a corresponding opinion inquiry task. A node in a DTM thus represents a *cognitive state* that has two ingredients - a working memory and a query in mind. An edge in a DTM now captures both the difference in memory and the difference in query between two connecting cognitive states.

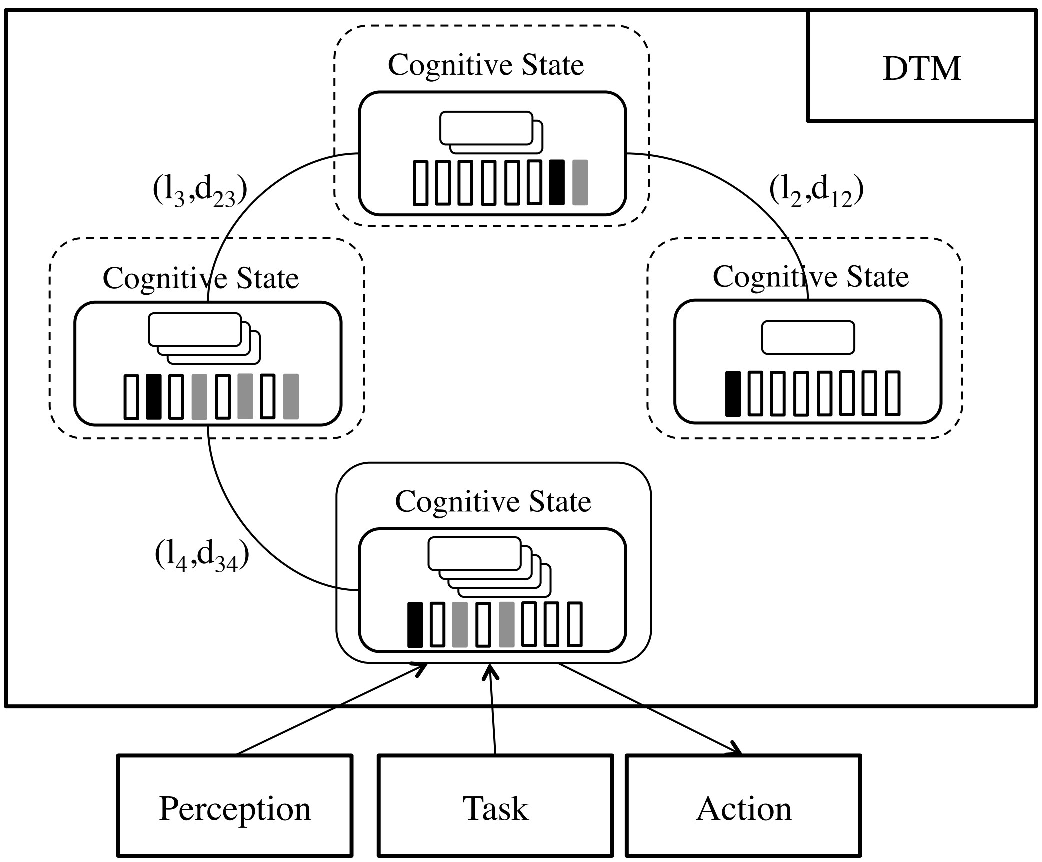


Figure 1.4: Double Transition Model

A DTM is capable of performing actions (here actions refer to cognitive activities); that is, inferring an opinion based on a query over the working memory in the current cognitive state. The action that is performed on a cognitive state is an inference activity that outputs an opinion in the form of probabilistic value - thus each cognitive state corresponds to one opinion. In essence, the structure of a DTM is a graph with a number of interconnected cognitive states, based on each state an opinion can be formed through inferencing.

We have illustrated a simple DTM in order to convey our basic ideas in developing a framework. However, there are three key challenges regarding the design of DTM.

The first challenge is whether a collection of episodic knowledge is a sufficient representation of knowledge. It has been widely accepted that humans store three different types of knowledge (Baddeley and Hitch, 1974; Coleman and Mizel, 1968): episodic knowledge, semantic knowledge and procedural knowledge. Episodic knowledge is the things we "remember" - the situations we have experienced, things we have seen and so forth. Semantic knowledge is the things we "know" - e.g. The earth is round. Procedural knowledge is about the "how" - e.g. The steps to make a cup of coffee. We will explain in Chapter 4 how we manage to simplify the problem by deriving semantic knowledge from the collection of episodic knowledge by conducting episodic-based reasoning and semantic-based reasoning. By doing so, we reduced the complexity of instantiating a framework.

The second challenge is whether the vector representation of a query is a sufficient representation of opinion requests from human. Humans can ask questions in a variety of forms such as yes-no question (e.g. Is today Thanksgiving?), alternative question (e.g. Is this building tall or short?), wh-questions (e.g., Who is the current president of United States?), and tag questions (e.g. Today is Thanksgiving, isn't it?). Therefore, we need to carefully evaluate whether a vector representation is sufficient to capture the semantic meaning in questions expressed by natural language. We present an intermediate representation (propositional logic) as a standard language for engineers to encode questions expressed by natural language, and then, we provide a method that automatically translate an opinion request in the form of propositional logic to opinion queries in the form of vectors.

The third challenge is the difficulty to identify the *current* cognitive state in use for deriving an opinion. It may seem obvious that the most recent memory or the long-term memory should be chosen, but we will show in Chapter 4.2 that our understanding of the structure and content of a working memory is limited. What may even be worse, due to the ambiguity of communication or by deliberate choice, the query in use may not be equivalent to the query of the interlocutor. Mainly, a DTM is designed in order to reflect the dynamics of the cognitive states in use by explicitly modeling stimulus (edges) that can trigger changes. Transitions between the cognitive states is one of the internalizations we wish to learn for our overall problem.

1.1.3 Phase II: Augmented Double Transition Model + Reinforcement Learning

In Phase I, we focus on the development of the Double Transition Model. A DTM can perform simple tasks such as forming an opinion given a query. Figure 1.5 shows how a DTM reacts to a dynamic environment. For consistency, we also refer to a perception activity as a task. As shown in the sequence on the top of the figure, l_1 to l_3 are perceived learning episodes in three consecutive steps. Tasks 4 and 5 are opinion formation tasks: For example, in task 4, an individual asks for an opinion on query q_1. The DTM then forms an opinion o_1 and sends it out. In the task sequence shown at the top of the figure, the rectangles colored as white (e.g. l_1) are received information including perceptions (represented by learning episodes) and task information (represented as queries). The black rectangles (e.g. o_1) are outgoing messages. We call tasks 4 and task 5 as non-episodic opinion formation tasks as they

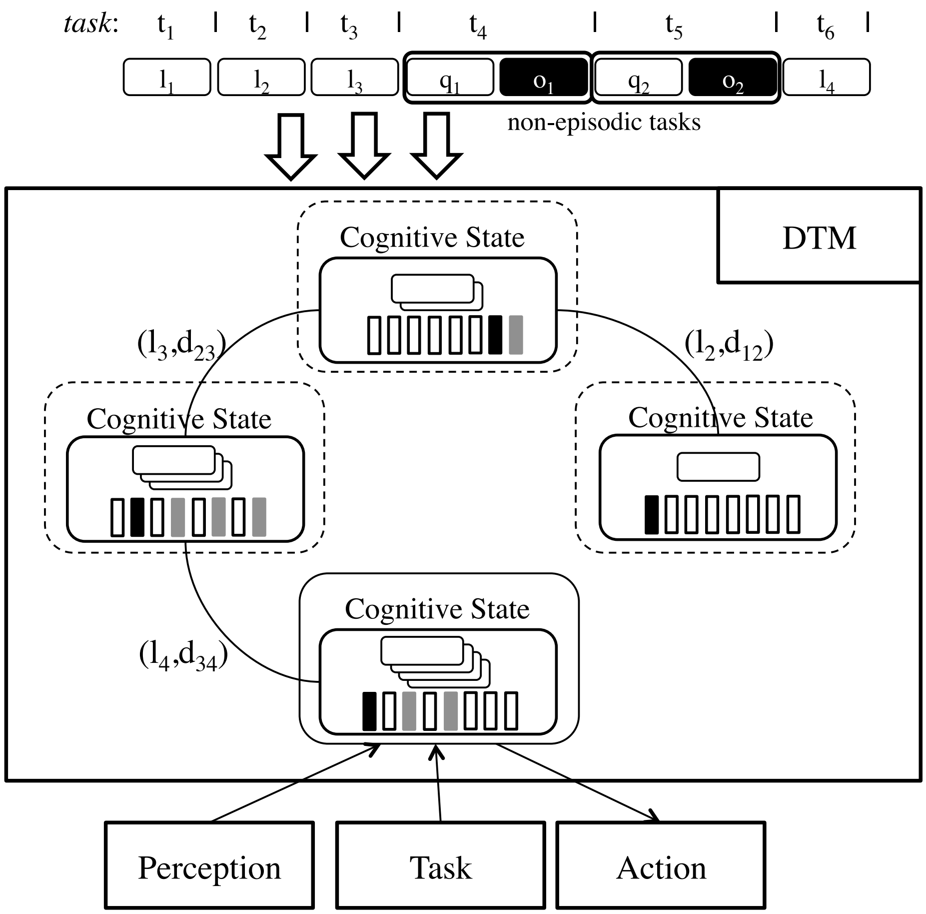


Figure 1.5: Behavior for a Double Transition Model

do not maintain an interactive session with the individual who asks for opinions. In real-world problems, asking a librarian's opinion can be considered as a typical non-episodic opinion formation task where an individual tends to be the domain expert. Non-episodic opinion formation tasks are common also when the problem itself is not controversial. *Episodic opinion formation tasks* are also common. For example, if two people are estimating the weight of a cow in a photo, they can go through a couple of iterations to describe their estimation as well as reasons.

In Phase II (covered in Chapters 6 and 7) our focus is to model episodic opinion formation tasks. We break down this task further with regards to whom an individual interacts with: non-dynamic one-to-one episodic opinion formation task[4] focuses on how to choose an action to fulfill his goal whereas his interlucator also has a particular goal to fulfill. A goal can simply be reaching opinion consensus, changing the other's opinion, or do not care; *dynamic one-to-one episodic opinion formation task*[5] is that an individual engages in opinion formation task with different people one at a time.

Non-Dynamic Episodic Opinion Formation Task

As shown in Figure 1.6, task 5 is an episodic opinion formation task. In this task, an individual receives a query for providing an opinion (q_2), and then he responds with opinion o_2. After he responds with his opinion, he receives feedback from the other agent: o_3 represents the other individual's opinion on the same query and l_4 is the other individual's explanation.

From the perspective of interaction, a non-dynamic episodic opinion formation task (Figure 1.6) includes three subtasks which differs from a non-episodic opinion formation task:

1. determines whether to accept a message sent by someone else.

2. determines whether to provide information to explain his opinion.

3. determines whether to adjust his own opinion.

In the first task, one may re-form his opinion (usually called opinion change) by accepting new information from someone else. In the second task, he makes a decision on whether to adjust his opinion by observing others' opinions. In the third

[4]As this thesis focuses on interactions between two entities, we simply consider this task to be a non-dynamic episodic opinion formation task.

[5]As this thesis focuses on interactions between two entities, we simply call this task a dynamic episodic opinion formation task.

task, the generation of information can be considered a by-product of the opinion itself. The importance of this by-product is that it becomes an external influence (actually a perception) to other people he communicates to.

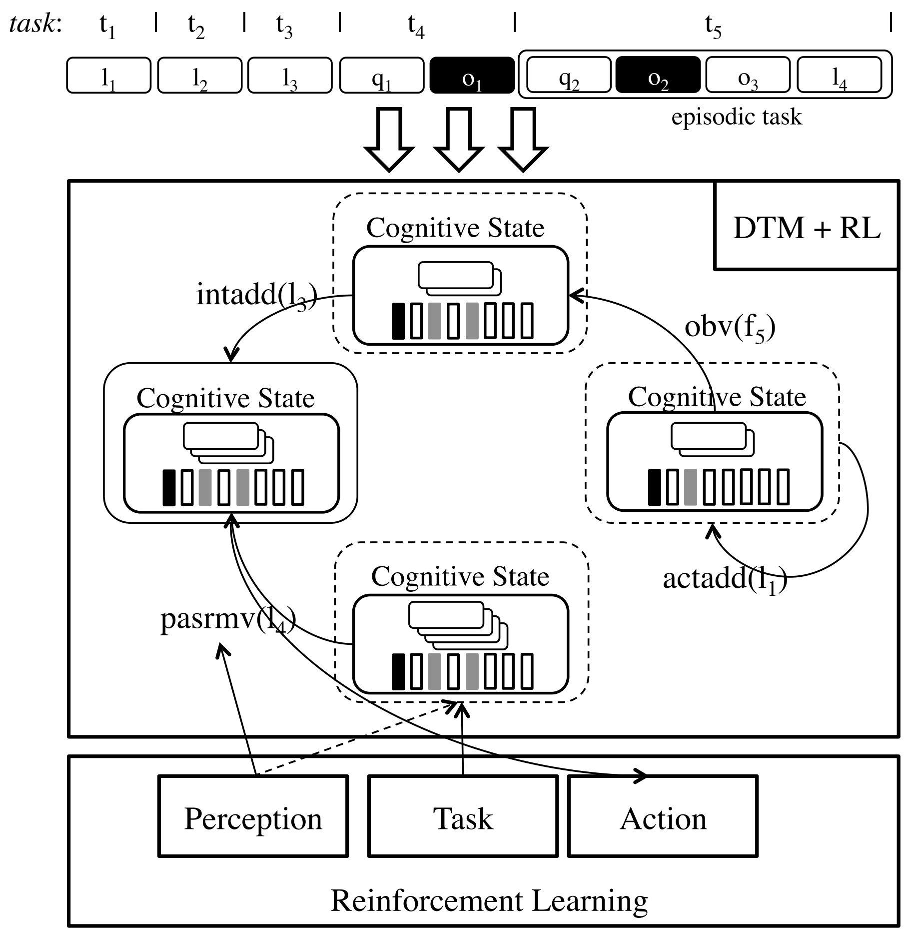


Figure 1.6: Non-Dynamic Episodic Opinion Formation Tasks

As we described earlier, the existing computational models solve each task separately. Each task is implemented independently of another task which makes it non-intuitive and hard to analyze. Most importantly, it is not coherent with the nature of human behavior. Ideally, when two people start discussing on one issue, they go through a couple of iteractions of opinion exchange with additional arguments. Then, they either reach a consensus or ends the conversation or exchange with a unresolvable opinion polarization.

We solve a non-dynamic episodic opinion formation task by defining it as a Markov decision problem. Each player makes a decision on the best action to take to maximize his/her utilities (values). An action can be to accept a message from the other, to send a chosen message to the other, to adjust his own opinion by recognizing/discarding a previously seen message, or just do nothing. The DTM model and the goal function of each player is sufficient to define all the quantities of a MDP.

Dynamic Episodic Opinion Formation Task

In Chapter 6, we describe our modeling approach. In Chapter 7 (see Figure 1.7), we describe how one player can learn how to perform well using Q-learning methods when he is not familar with the reactions from the other player. Q-learning also allows us to model dynamic episodic opinion formation tasks[6]; that is, one player engages in an opinion formation task with N different players one at a time. As shown in Figure 1.7, different colors in the environment sequence represent different individuals.

The framework of computational opinions is a reinforcment learning framework with double transition model as a core. Despite of its simplicity, it can model the entire sequence of opinion formation tasks.

1.2 Contribution

We provide a new outlook of the nature of opinion modeling. The key contribution to the framework of opinions is:

[6]We may also call it as dynamic one-to-one episodic opinion formation task.

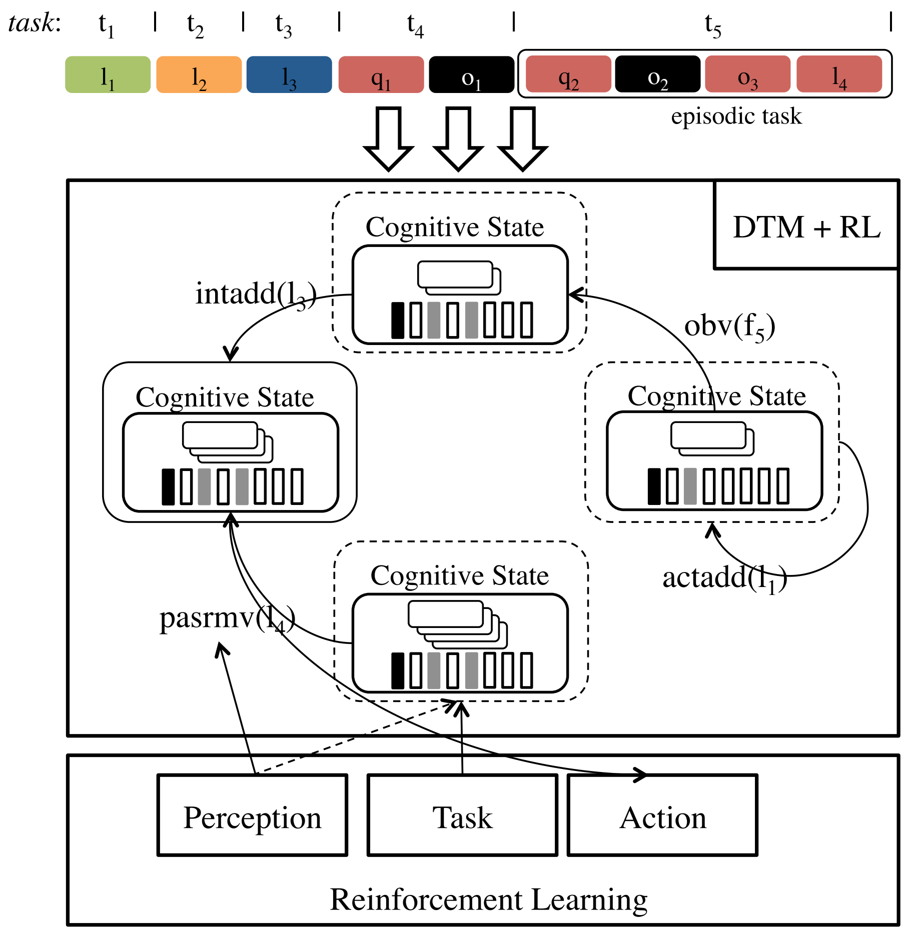


Figure 1.7: Dynamic Episodic Opinion Formation Task

- first framework that models knowledge basis, opinions, and interactive formation with external influence
- first framework in which the cognitive model is learned from feedback cycles between opinion formation and opinion change
- little requirement for knowledge engineering
  A double transition model is constructed on episodic knowledge in the form of a matrix and a history of queries being asked in the form of vectors.
- is domain-independent thus does not need to build one DTM for each opinion
- there is high transparancy as the framework is mathematically defined

Key contributions to *opinion theories* are:

- provides a platform to study the persistence of opinion change
- provides a platform with which to study various propositions of how and why humans differ in opinion formation
- provides a platform to study the relation between internalization of external influence and externalization of internal influence
- an opinion can be analyzed via knowledge structure and its content for other models

Key contributions to theories of rationality are:

- compatible with dual-process theory
- first framework that unifies some commonly-accecpted human heuristics

Key contributions to problem solving in opinion modeling are:

- defines and provides algorithms to solve a (non-)episodic opinion formation task
- defines and provides algorithms to solve a (non-)dynamic episodic opinion formation task
- defines and provides algorithms to solve a sequence of (non-)episodic (non-)dynamic opinion formation tasks
- provides solutions to convert questions expressed in natural language into queries in the form of vectors
- provides a lazy fusion method to construct a probabilistic model from data in the form of a matrix

1.3 Outline

This thesis presents a framework of computational opinions. We present our work in two phases. Phase I focuses on addressing the learning aspect of an opinion formation task.

- Chapter 2 surveys main social theories and theories of rational thinking in order to provide a general understanding of the main challenges these two fields have and the main tasks and problems they attempt to address. Our objective is to develop a computational framework that is general, compact, simple to build, and compatible with both social theories in opinions, and theories of bounded rationality.

- Chapter 3 provides background for the three technologies we use in building the framework: Makov decision process is used to model an opinion formation task, Bayesian knowledge bases are chosen as both knowledge representations and reasoning mechanisms, and additional reinforcement learning methods are used for model-free problem solving.

- Chapter 4 formally introduces Double Transition Model with justifications and methods presented in consequent sections.

- In Chapter 5 we provide some examples of how an opinion formation task is performed by a DTM and then we conduct an evaluation of DTM by modeling four human reasoning heuristics.

Phase II focuses on addressing the decision aspect of an opinion formation task. In particular, we present our work in defining and solving different types of common opinion formation tasks.

- In Chapter 6 we provide justifications on episodic opinion formation tasks and describe how to define a MDP based on DTMs and goal functions.

- As we assume full knowledge of environment dynamics in Chapter 6, we explore alternative reinforcement learning methods that can mitigate this problem in Chapter 7. Furthermore, we show that model-free methods, such as Q-learning, also allow us to model how an individual improves his strategy over repetitive exercises by performing the tasks with different individuals. In addition, we formally define multi-agent MDPs over a set of DTMs and goal functions.
- In Chapter 7, we also conduct a case study to demonstrate the computational power of the framework in studying a hypothetical problem of improper training.
- We conclude this thesis and describe a few important future works in Chapter 8.

1.4 Glossary

Opinion is personal beliefs that are derived rather than recalled; opinion is personal as it can be derived in diverse ways.

An *opinion change process* is a process to internalize an external influence.

An *opinion formation process* is a process to externalize internal cognitive processes.

A *non-episodic opinion formation task* consists of a single opinion formation process, in which there is no external influence in the task.

An *episodic opinion formation task* consists of a sequence of opinion formation and change processes.

A *dynamic episodic opinion formation task* consists of a sequence of opinion formation and change processes that can be further divided into episodic and non-episodic opinion formation tasks.

An *external influent* refers to one message from the world.

An *internal influent* refers to one message from oneself.

Chapter 2

Literature Review

The objective of this chapter is to survey the major theories in opinion modeling and theories in bounded rationality. In Section 1, we focus on three well-established social theories: Compliance-Identification-Internalization theory, which studies the persistence of opinion change from the perspective of social integration, Affect Control Theory which studies the role of sentiment in controlling opinions (sentiment), and Receive-Accept-Sampling Theory which places a central focus on knowledge in the process of opinion change (knowledge). Lastly, we will have a brief discussion on how a computational framework would be helpful for advancing the state-of-art of social theories.

Section 2 surveys research efforts in the rationality of thinking. By describing the main findings in this field, our goal is to convey the message about how humans solve problems differs from how computational methods and models solve problems. In the AI literature we differentiate the strong form of AI and the weak form of AI. Models that can simulate how humans think are considered strong AI and models that can simulate how human behave are considered weak AI (Russell et al., 2010). In general, weak AI focuses on methods and systems used in problem solving (Campbell et al., 2002; Marr, 1982) which has made significant progress in the last few decades. The progress of strong AI is much slower and it mainly focuses on replicating human cognitive capabilities such as learning (Bareiss et al., 1990), comprehension (Charniak, 1972), reasoning (Michalski, 1983), and planning (Lehman et al., 1996). Generally, the reasons for slow growth in AI is its difficulty with evaluation, and the availability and accessibility of datasets.

Despite the challanges in the AI field, deep learning (towards strong AI) has recently been gaining some attention. In 2009, the Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) method which overcame a long-standing problem in reconcurrent backpropogation, and won three International Conference on Document and Analysis (ICDAR) 2009 competitions in connected handwriting recognition without any prior knowledge about the three different languages to be learned. In 2012, the Google Brain team (Le et al., 2012) learned a 9-layered locally connected autoencoder from unlabeled images taken from YouTube videos to recognize higher-level concepts such as cats and human bodies. These two recent successes of deep learning have resurrected interests in multi-layer computational models especially in artificial neural networks (Haykin and Network, 2004).

This chapter does not survey all the known computational models mainly because there are few computational frameworks that simultaneously model knowledge, behavior, and the forward and backward feedbacks all in one framework. Instead, we describe relevant cognitive models in Chapters 4 and 5 when we describe DTM. We then describe relevant behavioral models in Chapters 6 and 7.

2.1 Theories on Opinions

Social influence on opinions has drawn a lot of attention in the research communities due to its long persistence and strong influence in both opinion change and behavior change.

Watts (Watts and Holt, 1979) argues that opinion change induced from active participation, such as writing or role playing, persists longer than from passive participation such as through reading. This is because active participation produces new information, leads to greater involvement, and makes participants continue to think about the event after the participation has ended. All of these activities result in better recall of the information. It has been also observed (Hoekstra, 1995) that public opinions are heavily influenced by the pressure conditions under which they were obtained. High pressure conditions such as punishment, reward, or external surveillance might induce unnecessary resistance or compliance to survey responses. Another reason why external influence on opinions is being heavily researched is that studies may provide unique insights on effective methods to change public opinion. This is important for both the media and the government, who often attempt to redirect public attitudes by providing accessible information and opinions from elites (Zaller, 1992).

2.1.1 Compliance-Identification-Internalization Model

The Compliance-Identification-Internalization Model (CII-Model), was developed by Kelman (Kelman, 1955, 1961, 1974) from the perspective of social integration with opinion communicators. According to Kelman, "this theoretical framework was particularly concerned with specifying the conditions under which changes induced by social influence attempts are temporary and superficial and, by contrast, those under which such changes are lasting and integrated into the person's belief and value systems". The CII model thus allows for differentiating three different processes of influence: compliance, identification, and internalization. Briefly speaking, *compliance* refers to the situation where external influence is accepted by an individual because he wants to gain specific rewards such as approval, avoid specific publishments, or disapproval by conforming, not because he believes in the content. Individuals seek to be compliant in terms of opinion values but the individuals' underlying beliefs may not be consistent. *Identification* refers to the situation where external influence is accepted by an individual because he wants to establish or maintain a satisfying self-defining relationship to another person or a group. He believes in the responses but their specific content is more or less irrelevant. *Internalization* refers to the situation where influence is accepted by an individual because both the content and the behavior itself are intrinsically rewarding.

Kelman conducted an empirical experiment in 1954, just prior to the announcement of the Supreme Court decision on desegregation in the public schools. They recorded communications in tapes that advocated maintaining some of the private Negro colleges to preserve local culture, history and tradition if the Supreme Court ruled that segregation was unconstitutional. They varied the source, degree of the communicator's power (e.g., means-control, attractiveness, and credibility). In addition, they also varied the conditions of salience and surveillance. Salience is maximized by issuing the questionaire immediately after the communication. Surveillance is maximized by designing the questionaire to strengthen that the questionaire is administered at the communicator's request. The experimental results confirmed the hypotheses of the CII model.

The CII model is one of the most important theories on social influence which inspired numerous studies in various domains ranging from educational exchange (Bailyn and Kelman, 1962), organizational commitment (Mowday et al., 1979), and to analyze contemporary online social networks such as Facebook (Cheung et al., 2011). The CII theory points out that the characteristics of an opinion communicator, the way an opinion gets conveyed, and internal mechanisms play important roles in determining the final influence of an opinion. Our rationale for an overarching computational framework is very much aligned with the motivation for CII theory, however, while CII theory focuses on conditions and various processes under which an influence is accepted, the theory does not address how such influence may propogate within communities. For example, if an influence is internalized by an individual, how would this individual become a communicator to others and what will be the content this individual expresses? The extension of CII theory to the processes of influence acceptance and process of influence generation would be particularly valuable in the design of a computational frameworks.

2.1.2 Affect Control Theory

While Kelman's CII theory is concerned with various extents of internalization of external influence, Affect Control Theory (ACT), (Berger and Zelditch, 2002; Heise, 1979; Robinson et al., 2006; Smith-Lovin, 1987), is concerned with the interplay between affect and behavior. According to Heise, the ACT theory has three basic axioms:

1. Individuals create events to confirm the sentiments that they have about themselves and others in the current situation.

2. If events don't work to maintain sentiments then individuals re-identify themselves and others.

3. In the process of building events to confirm sentiments, individuals perform the social roles that operate society–the principle of affective rationality.

The mathematical model of ACT theory (Heise, 1979; Robinson et al., 2006; Rogalin et al., 2007) utilizes a few tools such as sentiment measurements, impression-formation equations, and mathematical minimization procedures to derive an EPA (Evaluation, Potency, and Activity) profile. Evaluation, Potency and Activity were the most significant three dimensions of affect identified by Osgood et al. (Osgood et al., 1975) after conducted a cross-national project from 1950 to 1960 in order to test the hypothesis that human beings utilize similar descriptive frameworks in allocating affective meaning of concepts. Test subjects from two dozen societies were presented with a list of common concepts (e.g. father, water) and were asked to respond to each concept with a modifier (e.g., beautiful, scary). An atlas of affective meaning for some 600 concepts was developed through a variety of cross-national investigations on measuring affective associations of concepts. Based on statistical analysis the three factors which were identifiable in nearly every language and cultural community were *Evaluation* (e.g. good, bad), *Potency* (e.g. strong, weak), and *Activity* (e.g. active, passive). Described by Berger et al. (Berger and Zelditch, 2002), Evaluation refers to a sense of approval or disapproval that can elaborate into judgements of morality, aesthetics, functionality, hedonism, or other standards, Potency relates to an entity's impact and might elaborate into assessments of physical magnitude, strength, forcefulness, social power, expansiveness, and the like, and Activity indexes an entity's spontaneity which can elaborate into judgments of animation, speed, perceptual stimulation, age, propensity to be an agent, and so on.

While CII theory focuses on identifying three processes each associated with a different degree of persistence in opinion change, ACT highlights the possible reinforcement effect between affect and behavior. According to the 2^{nd} axiom, sentiment towards another individual not only results in a particular degree of persistence (predictable using the EPA factors) for that scenario, but also results in changes to internal mechanisms to fit how the persistence of opinion change will be for the future scenarios.

2.1.3 Receive-Accept-Sampling Model

An alternative model for studying political communication and public opinion in general is the Receive-Accept-Sampling Model (RAS) developed by Zaller (Zaller, 1992). The CII model differentiates processes of opinion change based on the relationship between behavior compliance and the consequent compliance with an internal value system, however, the question of how influence may conflict or is aligned with one's internal value system was not formally elaborated in this model. The RAS model characterizes opinion formation process from the perspective of information processing: the opinions expressed by individuals are heavily influenced by their degree of political awareness (receive), consistency between the messages and their predispositions (accept), and the priorities of messages by the time opinions were formed (sampling). As pointed out by Zaller, politically more aware individuals are more likely to receive elite message. Due to their exposure to multiple and often conflicting messages, people are less likely to accept messages that are inconsistent with their prior attitudes. Less aware individuals receive fewer messages, but are more likely to accept them even if they are conflicting.

Compared to the CII model and ACT model, the cognition-oriented RAS model studies from how cognitive engagement influences the persistence of opinion change (accept), how cognition is involved in evaluating information (accept), and how information is selected in deriving an opinion (sampling).

2.1.4 Computational Models

Recently psychologists have incorporated techniques from information systems to model human behavior. For example, Chwelos et al. (Chwelos et al., 2001) proposed a model for investigating a purchasing manager's intent for adopting inter-organizational systems such as an electronic data interchange. Early work by Price and Mueller (Curry et al., 1985) models the causal relationship between the employment turnover of nurses and the turnover rate's determinants. Welner (Schmidt and Weiner, 1988) proposed a cognitive, emotion, action model to analyze people's decision on help-giving. In another example, Sheth (Sheth, 1973) developed a model that integrated psychology, marketing, and information systems to predict online consumer behavior. More specifically, due to an online consumer's dual nature as a traditional shopper and a computer user, the Technology Acceptance Model (TAM) (Davis et al., 1989) was applied to investigate their intent at using the technology of the web store, where TAM itself is a model used to predict the use of information systems.

All these behavior models take the form of a causal model that predicts behavior based on its correlation with internal determinants such as intent, attitude, belief, and emotion, as well as with external determinants such as resources, usefulness, and ease of use. These models were built on careful analysis of real-life data and verified through empirical studies, however, they do not account for the dynamism of people's behavioral change and opinion change. Moreover, in each of the models, the variables and causal relationships are fixed so they are limited to only their respective domain applications.

2.1.5 Discussions

The general issues reagarding social theories are that they are typically domain-specific, meaning that validating social theories in other domains take tremendous time and labor. These studies are often pursued by scholars from a variety of disciplines, and so constructively comparing results across different experimental designs, executions, and data is a challenge. The lack of a computational framework has limited scholars for the problems they can explore and the hypotheses they can verify. For example, the issue of conveyed information is clearly important in the process of opinion formation. According to LeDuc (Leduc, 2002), the more the issue is linked with one's political identity (or inner belief) or the more knowledge the voters have about the issue, the more predictable the voting will be, and the faster voters will make up their mind. On the other hand, the newer the issue is (e.g., maybe the issue is seldom debated publicly), the more volatile the voting is, and the longer it takes to form decisions. Current studies are pursued mainly via computing correlations between information and opinions at the macro-level (population dynamics), rather than be capable of exploring opinion dynamics on the micro-level (individual opinion dynamics). Another problem is that the assumptions of theories and experiments can be inconsistent with each other. For example, analysis of public opinion has also shown that public opinions are unstable and are even affected temporarily by the framing of the survey questions (Kelman, 1961), order of questions on the survey, or the choice of survey questions immediately before the response (Zaller, 1992).

2.2 Theories on Rationality

In the work of Gigerenzer and Brighton (Gigerenzer and Brighton, 2009), they listed some interesting examples of how animals use smart heuristics to solve adaptive problems:

> "To measure the area of a candidate nest cavity, a narrow crack in a rock, an ant has no yardstick but a rule of thumb: Run around on an irregular path for a fixed period while laying down a pheromone trail, and then leave. Return, move around on a different irregular path, and estimate the size of the cavity by the frequency of encountering the old trail. This heuristic is reported to be remarkably precise: Nests half the area of others yielded reencounter frequencies 1.96 times greater (Mugford et al., 2001). To choose a mate, a peahen similarly uses a heuristic: Rather than investigating all peacocks posing and displaying in a lek eager to get her attention or weighting and adding all male features to calculate the one with the highest expected utility, she investigates only three or four, and chooses the one with the largest number of eyespots (Petrie and Halliday, 1994)."

Simon (Herbet, 1955) also suggests that humans may adopt heuristics to make a decision even though we have demonstrated superior cognitive capabilities compared to animals. In particular, Simon pointed out that "humans experience limits in formulating and solving complex problems and in processing (receiving, storing, retrieving, transmitting) information". Simon's theory of Bounded Rationality has inspired research in domains from Economics (Duhaime and Schwenk, 1985; Schwenk, 2006), Psychology (Kahneman et al., 1982), Political Science (Cohen et al., 1972), and AI (Brighton, 2006), literally all fields that have something to do with humans.

In general, researchers are interested in studying how humans conduct congitive activities under the situation of bounded rationality. One aspect for approaching this question is to study the relationship between cognitive processes and their performance. There are two main streams of research, one streams focusing on cognitive biases and the other one focusing on effective heuristics in reasoning.

Kahneman (Daniel, 2011) and Tversky (Kahneman et al., 1982; Tversky, 1972) introduced the notion of cognitive biases to describe the mistakes individuals constantly and consistently make. For example, the ambiguity effect is a tendency to avoid options for which missing information makes the probability seem "unknown". Furthermore, their studies sugggested that human cognitive activities may not be in accord with existing mathematical models (e.g. statistical models). For instance, they conducted experiments showing that humans may reach a wrong conclusion when they ignore prior probabilities (Kahneman and Tversky, 1972). As Kahneman explained in his Nobel Memorial Lecture (Kahneman, 2002): "Our research attempted to obtain a map of bounded rationality, by exploring the systematic biases that separate the beliefs that people have and the choices they make from the optimal beliefs and choices assumed in rational-agent models.".

Gigerenzer and his colleages approach the problem from another viewpoint; that is, instead of identifying road blocks that prevent one from making an optimal choice (like a rational agent would do), the identified approaches/patterns that humans have successfully applied to make good choices. In the past decades Gerd Gigenrenzer and his team focused on a collection of fast and frugal heuristics (Czerlinski et al., 1999; Gigerenzer and Brighton, 2009; Gigerenzer and Gaissmaier, 2011; Gigerenzer and Goldstein, 1999; Martignon and Hoffrage, 2002) that have been shown to outperform complex mathematical models such as linear regression and decision trees.

Even though Kahneman and Gigerenzer are looking at two different sides of the same coin, it is hard to integrate their findings. Their different view points grew out of these two streams. Kahneman and his team have made implicit assumptions that humans do optimize while Gigenrenzer argues that the assumptions of sub-optimality deviates from Simon's original idea (Gigerenzer and Gaissmaier, 2011). Another consideration is that brain mechanisms in cognitive activities still remain a mystery to all of us. We are still puzzled by simplier questions such as how does a genius' brain differs from an ordinary's brain (Simonton, 2012). Knowing how a brain works in a problem-solving or a decision-making process is a more challenging problem. Human brains are still an unknown black box. The common observables are human actions and communication. For us to try understanding cognitive activities, however, there are many unresolved issues that provent us from even analyzing human communication and behavior, specifically human deceptions hide true human intent from observers, opinion and behavior shaped by the environment, exogenous influence from other peers and nowadays from multimedia that can change one's opinions without one even noticing it (Baron, S., 2005; Esser, 1998; Mason et al., 2007).

It is evident that a computational model may help test the validity of these two perspectives, explore conditions under which any of these two holds or fails, and, more importantly, provide capabilities to automate all these tests. In general, a computational model is necessary for validating different theoretical accounts and empirical findings, and it also serves as a framework for scholars to communicate and replicate each other's work.

To the best of our knowledge, there is no such computational model that can achieve the required integration. The prevalent computational models assume some forms of omniscience (have knowledge of all data needed) and some forms of omnipotence (have all the computational resources and can always compute them correctly), however, the properties of omniscience and omnipotence of these computational models are inconsistent with the nature of human cognitive processes that typically have partially observable information, have limitations on computational complexities, and can have errors in perception and reasoning. Works in the areas of Fuzzy Logic (Klir and Yuan, 1995), Imprecise Probability (Dempster, 1967), and numerous other methods of missing value estimations (Schafer and Graham, 2002) are often treated as methods to remedy situations where things are not as perfect as expected.

Another challenge for a computational model is to be capable of embodying different types of individual differences relevant to human cognition. In studying factors that drive opinion change, Kelman (Kelman, 1961) has stated that knowing the direction of an individual's response or the distribution of responses in the population are far from the information needed to understand these opinions. We end this chapter with the three types of information considered important by Kelman in understanding opinion formation and opinion change:

> *"We need information that will allow us to make some inferences about the characteristics of the observed opinions - their intensity, their salience, the level of commitment that they imply. We need information about the motivational bases of these opinions-about the functions that they fulfill for the individual and the motivational systems in which they are embedded. We need information about the cognitive links of the opinions-the amount and the nature of information that suppports them, the specific expectatations that support them."*

Chapter 3

Background

This chapter provides background material on Markov decision processes, model-free reinforcement learning methods, and Bayesian knowledge bases. Bayesian Knowledge Bases is our choice for knowledge representation of memory as well as our choice for modeling human reasoning (in Chapter 4). Markov decision processes is our choice to model both non-episodic opinion formation tasks and episodic opinion formation tasks between two people (in Chapter 6). Model-free reinforcement learning methods is our choice to generalized the framework for modeling non-episodic and episodic opinion formation tasks between two entities (in Chapter 7).

3.1 Markov Decision Process

3.1.1 Notations

Definition 1 (Discrete-Time Finite Markov Decision Process)**.** A *discrete-time finite markov decision process* (MDP) is a 4-tuple (S, A, P, R) (Sutton and Barto, 1998), where

- S is a finite set of states.
- A is a finite set of actions.

- P is function from $S \times A \times S$ into $[0,1]$, the closed unit interval, such that $\sum_{s' \in S} P(s,a,s') = 1$ for all $a \in A$ and $s \in S$.

- R is a function from $S \times A \times S$ into $\mathcal{R}$, the collection of all real numbers.

In the above definition, $P(s,a,s')$ is the probability that the next state is s', given the action a and the current state s. In other words, $P(s,a,s') = Pr(s' \mid a, s)$. We shall also denote $P(s,a,s')$ by $P_a(s,s')$. Moreover, $R(s,a,s')$, also written as $R_a(s,s')$, is the value of the reward received after transition to state s' from state s by taking action a.[1]

In what follows, t will denote the time step, s_t the state at t, and a_t the action at t.

Markov Property is the key assumption of a MDP. When we consider how the environment responsed to an action, say a, the chances for various outcomes may depend on the entire history, described in (Sutton and Barto, 1998) as

$$Pr\{s_{t+1} = s', r_{t+1} = r | s_t, a_t, r_t, s_{t-1}, a_{t-1}, \cdots, r_1, s_0, a_0\}.$$

for all s', r, and all possible values of the past events: $s_t, a_t, r_t, \cdots, r_1, s_0, a_0$. The Markov property is a property that the environment's next response to an action only depends on the current environment state and action and is independent of the entire history. In other words, Markov property describes the environment with *one-step dynamics*, as follows

$$Pr\{s_{t+1} = s', r_{t+1} = r | s_t, a_t\}$$

[1] This reward function allows the reward to depend on the action and also the outcome. A simpler but also commonly used reward function is $R_a(s)$, where the immediate reward depends only on the action but not the outcome.

If the environment satisfies Markov property, the next state (transition probability) as well as the reward (reward function) then can be predicted based on the current state as well as the action taken in that state.

Another important assumption made in a MDP is the **full observability** of the world meaning that an agent knows which states they are in with complete certainty.

Definition 2 (Infinite Horizon and Finite Horizon)**.** Infinite horizon and finite horizon is a fundamental property for a sequential decision problem. In a *finite horizon* decision problem, the agent is only allowed to take a finite number of actions. However in an *infinite horizon* decision problem, the number of actions an agent can take is unlimited.

This thesis focuses on infinite horizon MDPs, thus we introduce the definitions (e.g. policy, utility, and basic solutions to derive optimal policy) assuming that they are for the infinite horizon problems. However since an agent can behave dramatially under different problem setting, we will provide some brief discussion on the impacts of this property.

Definition 3 (Policy)**.** Policy (also be called as strategy or plan) is the solution to answer what action an agent should do at a state. Formally, we denote a policy by π, and $\pi(s)$ is the action recommended by the policy π for state s. With a complete specification of policy, an agent will always know what to do for any state.

Each time a policy gets executed starting from the initial state, the stochastic nature of environment will leads an agent to be in a different environment history. Therefore, the performance of a policy can be measured by the *expected* value of all possible environment histories generated by this policy. We use π^* to denote an optimal policy which has highest expected value.

A policy derived for an infinite horizon problem is *stationary* in the sense that the selection of an action at a particular time does not differ from the selection that would be made at any other time. If a problem has a finite horizon (the game is over in a finite number of steps), the optimal action in a given state could change over time (policy under finite horizon is *nonstationary*). For example, if an agent has unlimited time to find a route to exit a maze, it can take its time to well explore the maze. In the case of a finite horizon situation, an agent near a wall at a later time would head directly to the exit while it may wander around the wall when the game just begain. More detailed theoretical discussions on various classes of policies ranging from randomized, history-dependent policies to stationary deterministic policies can be found in (Puterman, 2009).

Definition 4 (Value Function)**.** Let $V^{\pi}(s)$ denote the expected value (also be called utility) for executing the policy π starting from initial state s_0. We can derive this value by calculating the expected sum of all the rewards collected throughout all possible environment histories (Russell et al., 2010)

$$V^{\pi}(s_0) = E[\sum_{t=0}^{\infty} \gamma^t R(s_t)]$$

where γ is a discount factor and the expectation is with respect to the probability distribution over state sequence determined by s and π.

A *optimal policy* is thus the one that can yield highest value

$$\pi^*_{s_0} = argmax_{\pi} V^{\pi}(s_0)$$

As we consider infinite horizon MDPs where policies are stationary, we have

$$\pi^*_{s_0} = \pi^*_{s_1} = \cdots = \pi^*_{s_t} \text{ for any } t$$

Thus, we use a general notation[2] π_s that

$$\pi_s^* = argmax_\pi V^\pi(s)$$

The value of a state given a policy is derived based on *expected total discounted reward criteron* which is the most widely used criteron. The discount factor γ is both mathematically attractive and semantically intuitive. With rewards being discounted over time, the value of an infinite state sequence is finite. It has been shown that the value is in fact has a upper bound with the maximal reward (if $\gamma < 1$) (Russell et al., 2010). Facing a decision problem, the discount factor represents the agent's take on the importance of the near future. If the discount factor is 0, the agent is considered short-sighted as it only cares about the return of the current action. The larger the discount factor is, the more far-sighted an agent is.

Expected total reward criteron and *Expected average reward criteron* are two other alternative criteria to calculate the value of a policy. (Detailed analyses can be found in (Puterman, 2009)), but in this thesis we will stick with the discounted reward.

The value function allows an agent to act optimally by selecting an action that will yield the highest expected values of the next state

$$\pi^*(s) = argmax_{a \in A(s)} \sum_{s'} P_a(s, s') V^{\pi^*}(s')$$

where $A(s)$ is the state-action function to determine the scope of actions an agent is allowed to take in state s.

Definition 5 (Partially Observable Markov Decision Process)**.** When full knowledge of the environment is not available (full observability assumption does not hold), for example a player does not know which state it is in, then this player can no longer

[2]This no longer holds for finite horizon problems as the policies are non-stationary.

execute policy $\pi(s)$ even if it has one. *Partially observable Markov decision process* (abbrev. POMDP) is a generalization of a MDP that address sequential decision problems within partially observable environment.

POMDPs have key elements as a MDP but has an additional observation space ω and an observation function $O_a(s', o)$. Formally, a discrete-time finite partially observable markov decision process (abbrv. POMDP) is defined as a 6-tuple $(S, A, O, P_a(s, s'), R_a(s, s'), O_a(s', o))$ (Kaelbling et al., 1998; Russell et al., 2010), where

- S is a finite set of states $S = s_1, s_2, \cdots, s_n$.
- A is a finite set of actions $A = a_1, a_2, \cdots, a_m$.
- O is a finite set of observations that an agent receive.
- $P_a(s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$ is the probability that action a in state s at time t will lead to state s' at time $t + 1$.
- $R_a(s) = E(r_{t+1} | s_t = s, a_t = a)$ is the expected value of reward by taking action a at state s.[3]
- $O_a(s', o) = Pr(o_{t+1} = o | s_{t+1} = s', a_t = a)$ that $o \in O$ is the probability of making observation o of the actual landed state s' by taking action a.

If an agent has obtained full knowledge of the environment, then we have $O = S$ (Braziunas, 2003).

3.1.2 Algorithms

This section describes algorithms to find optimal policies. We will cover two popular classes of algorithms: *value iteration* and *policy iteration* with their varients. Consid-

[3]Note that the reward function specified for POMDP differs from the one specified for a MDP. For a POMDP, we consider a reward to only depend on the action but not its outcome.

ering an optimal policy π^*, we will have the following relationship (Sutton and Barto, 1998)

$$V^{\pi^*}(s) = \max_a \sum_{s'} P_a(s, s')[R_a(s, s') + \gamma V^*(s')]$$

In fact, this is the Bellman optimality equation (Bellman, 1956). As the other quantities are all specifieid in a MDP already, we will have thus n unknowns (states) and n equations. However, each equation has a max operator thus we need to solve n nonlinear equations. The value iteration algorithm uses an interative approach that is guaranteed to converge (proof of convergence can be found at (Puterman, 2009)). The basic idea is as follows: the algorithm starts with initial values for states (can be either arbitrary or be all zero). Then, the algorithm updates the value of each state by turning the Bellman optimality equation into a Bellman update rule shown below. When the value change for states is smaller than the pre-defined constant Θ (the final solution is thus referred to as Θ-optimal), the algorithm exits. The value iteration algorithm is shown in Figure 3.1.

$$V(s) \leftarrow \max_a \sum_{s'} P_a(s, s')[R_a(s, s') + \gamma V(s')]$$

The max operator in the value interation algorithm makes it harder to derive the exact solution efficiently. The *policy iteration* algorithm effectively tackles this problem when we already have a policy in mind. The Bellman equation for a policy π_i is mapped to

$$V^{\pi_i}(s) = \sum_{s'} P_{\pi_i(s)}(s, s')[R_{\pi_i(s)}(s, s') + \gamma V^{\pi_i}(s')]$$

Note that this equation no longer has the max operator and thus the exact solution for all state values can be derived by solving system of linear equations. The policy

Input: States plus terminal state S^+
Input: Transition probability table $P^a_{ss'}$
Input: Reward table $R^a_{ss'}$
Input: Discount factor γ
Input: Small positive constant Θ
Output: Deterministic policy π such that
$\pi(s) = \arg\max_a \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V(s')]$

var vector $V \leftarrow initToZero(S^+)$;
repeat
    var $\Delta \leftarrow 0$;
    foreach $s \in S$ **do**
        var $v \leftarrow V(s)$;
        $V(s) \leftarrow \max_a \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V(s')]$;
        $\Delta \leftarrow \max(\Delta, |v - V(s)|)$;
until $\Delta < \Theta$;

Figure 3.1: Value Iteration (Sutton and Barto, 1998)

iteration algorithm consists of two steps, beginning from some initial policy π_0 (Russell et al., 2010).

- **Policy evaluation**: given a policy π_i, calculate $V_i = V^{\pi_i}$ which is the value for each state if the policy π_i is executed.
- **Policy improvement**: calculate a new policy π_{i+1} using one-step look-ahead based on V_i.
- **Stopping rule**: the algorithm terminates if the policy improvement step does not result in changes in values.

Even though we can compute the exact value of all states given a known policy, it could still require a computational complexity $O(n^3)$ if standard linear algebra methods are used. The high computational complexity is problematic for a MDP with a large number of states. Therefore, instead of calculating exact values, we can use an iterative policy evaluation to reach a good approximation. The policy iteration algorithm (with iterative policy evaluation) is shown in Figure 3.2.

Input: States plus terminal state S^+
Input: Transition probability table $P^a_{ss'}$
Input: Reward table $R^a_{ss'}$
Input: Discount factor γ
Input: Small positive constant Θ
Output: Deterministic policy π such that
$\pi(s) = \arg\max_a \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V(s')]$

1. Initialization;
var $V(s) \in \mathbf{R}$ and var $\pi(s) \in A(s)$ arbitrarily for all $s \in S$;

2. Policy Evaluation;
repeat
    var $\Delta \leftarrow 0$;
    foreach $s \in S$ **do**
        var $v \leftarrow V(s)$;
        $V(s) \leftarrow \sum_{s'} P^{\pi(s)}_{ss'}[R^{\pi(s)}_{ss'} + \gamma V(s')]$;
        $\Delta \leftarrow \max(\Delta, |v - V(s)|)$;
until $\Delta < \Theta$;

3. Policy Improvement;
policy-stable $\leftarrow$ true;
foreach $s \in S$ **do**
    var $b \leftarrow \pi(s)$;
    $\pi(s) \leftarrow \mathrm{argmax}_a \sum_{s'} P^{\pi(s)}_{ss'}[R^{\pi(s)}_{ss'} + \gamma V(s')]$;
    if $b \neq \pi(s)$ **then** policy-state $\leftarrow$ false
if *policy-stable* **then**
    stop;
else
    go to 2;

Figure 3.2: Policy Iteration using Iterative Policy Evaluation (Sutton and Barto, 1998)

3.2 Reinforcement Learning

In fact, the process of learning within a MDP environment for the purposes of determining optimal behavior is reinforcement learning. According to (Sutton and Barto, 1998),

> "*Reinforcement learning is learning what to do—how to map situations to actions—so as to maximize a numerical reward signal. The learner*

is not told which actions to take, as in most forms of machine learning, but instead must discover which actions yield the most reward by trying them."

This definition of reinforcement learning illustrates the fundamental difference between reinforcement learning and supervised learning - reinforcement learning focuses on learning through interaction while supervised learning focuses on learning directly from collections of environment histories. In other worlds, reinforcement learning exploits the structure of the environment while supervised learning does not.

MDPs are important to the theory of reinforcement learning for the situation where the decision problem satisfies Markov property. In the previous section, we introduced two basic methods (value iteration Algorithm 3.1 and policy iteration Algorithm 3.2) to derive optimal policies when the full knowledge of environment is available. An agent may still have full observation of the environment that knowing the current state it is in with full certainty, however the agent may not know the *dyanmics of environment* which is captured in the transition probabilities $P_a(s, s')$ and immediate rewards $R_a(s, s')$ (also called sense). Sutton (Sutton and Barto, 1998) has described an intuitive example summarized here as follows

> Blackjack is a popular casino card game. The objective is to obtain cards with a sum as great as possible and not exceeds 21. The game between a player against a dealer begins with two cards dealt to both dealer and player. One of the dealer's card is faceup and the other card is facedown. If the player does not have exact 21 whn the game begins, the player then can require additional cards until either he/she stops or goes bust (sum exceeds 21). Once the player stops, the dealer can require additional cards until either he/she stops or goes bust. Whenever either the player or the

dealer goes bust, he loses the game. Otherwise, whoever has a larger final sum wins.

The game can be formulated as a MDP. The decision a player faces depends on three variables: his current sum of cards (11-21), the dealer's faceup card (ace-10), and whether or not he holds a usable ace[4].

In the blackjack problem, the player has full knowledge of the environment: he can observe the three variables of the environment with full certainty, and he knows what are the actions he is allowed to take at each state. What he may not know is $P_a(s, s')$ and $R_a(s, s')$ as it is hard to determine the cards the dealer may get under different situations and to determine the expected reward that follows. Even if a good estimation can be obtained, the situation may be too complex to be fully analyzed.

We introduce two classes of reinforcement learning methods that approximate value functions when full knowledge of the enviroment is not available: one class of methods is called *Monte Carlo methods* and another class of methods is called *temporal difference methods*. Both methods learn from *simulated experience*. In general, monte carlo methods estimate value functions on an episode-basis while temporal difference methods estimate value functions on a step-basis.

Considering a collection of simulated episode experiences by executing a policy π, a Monte Carlo method can learn a state-value function V_s^π based on the return of visiting state s in each episode. An every-visit MC method estimates V_s^π as the average of the returns following all the visits to s in all experience and a first-vist MC method estimates V_s^π as the average of the return following the first visit to s in

[4]The most detailed description of the blackjack game can be found in (Sutton and Barto, 1998), p.112-114

1. Initialization;
var $\pi \leftarrow$ policy to be evaluated;
var $V \leftarrow$ an arbitrary state-value function;
var $Returns(s) \leftarrow$ an empty list, for all $s \in S$;

repeat

  (a) Generate an episode using π;
  (b) For each state s appearing in the episode:;
  $R \leftarrow$ return following the first occurrence of s;
  Append R to $Returns(s)$;
  $V(s) \leftarrow average(Returns(s))$;

until;

Figure 3.3: First-visit MC Method for Estimating V^{π}
(Sutton and Barto, 1998)

all experience (Sutton and Barto, 1998). The procedure for first-visit MC methods is shown in Figure 3.3. The update equation for first-visit α-constant MC method is

$$V(s_t) \leftarrow V(s_t) + \alpha[R_t - V(s_t)]$$

where R_t is the total return after time t within this episode.

The simplest temporal difference method $TD(0)$ is

$$V(s_t) \leftarrow V(s_t)\alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$$

where r_{t+1} is the immediate return after visiting state s at time t. These two simple equations illustrate the essence of how MC methods and TD methods differ. The procedure for TD(0) is shown in Figure 3.4.

Q-learning is the most famous off-policy TD method developed by Watkins (Watkins, 1989). A one-step Q-learning methods is as the following

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a)]$$

```
1. Initialization;
var V(s) initialized arbitrarily;
var π is the policy to be evaluated;
repeat
    foreach episode do
        Initialize s;
        repeat
            foreach step in the episode do
                a ← action given by π for s;
                Take action a; observe reward r and next state s′;
                V(s) ← V(s) + α[r + γV(s′) − V(s)];
                s ← s′;
        until until s is terminal;
until;
```

Figure 3.4: Temporal Difference Method to Estimate V^{π}
(Sutton and Barto, 1998) p. 135

Q-learning estimate an action-value function rather than a state-value function. The estimation has been shown to converge with probability 1 to the Q^* (Sutton and Barto, 1998). The procedure of Q-learning for shown in Figure 3.5.

Definition 6 (Action-Value Functions)**.** We consider action value methods in making action selection decisions. The simplest action selection rule is to select the action with highest estimated action value. The action chosen using this rule is called greedy action, as

$$Q_t(a^*) = \max_a Q_t(a)$$

This method exploits current knowledge of the environment without considering inferior actions which generate a higher value in the future. Therefore, ϵ-greedy action selection rule selects the greedy action for most of the time but selects a random action with a probability of ϵ.

An ϵ-greedy action selection rule has two potential issues: 1) it is likely to choose

1. Initialization;
var $Q(s,a)$ initialized arbitrarily;
repeat
    foreach *episode* **do**
        Initialize s;
        repeat
            foreach *step in the episode* **do**
                Choose $a \leftarrow$ from s using policy derived from Q (e.g, $\epsilon - greedy$);
                Take action a; observe reward r and next state s';
                $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$;
                $s \leftarrow s'$;
        until *until s is terminal*;
until;

Figure 3.5: One-step Q-learning Method Estimate $Q(s,a)$
(Sutton and Barto, 1998) p. 149

worst actions due to the random selection, 2) it can take a while to pick the optimal action when the size of actions is large. The *softmax action selection rule* effectively addresses these issues is to vary the action probabilities as a graded function of estimated values. The most common softmax methods uses a Gibbs, or Boltzmann distribution. It chooses an action a at time t with probability (Sutton and Barto, 1998)

$$\frac{e^{Q_t(a)}/\tau}{\sum_{b=1}^{n} e^{Q_t(b)}/\tau}$$

where τ is a positive parameter called the temperature.

3.3 Bayesian Knowledge Bases

This section introduces Bayesian Knowledge Bases - a probabilistic knowledge represenation with capabilities of answering probabilistic questions. Bayesian Knowledge Bases encodes domain knowledge in a graphical representation by specifying “If-then” rules between states of variables. Bayesian Knowledge Bases allow knowledge to be incomplete - conditional probabilities between two random variables need not to be fully specified, allows cycles in the knowledge. In addition, the fusion algorithm provides capability to merge multiple BKBs into one BKB but still with individual source fragments identifiable.

We first describe the graphical representation of BKB (Santos Jr and Santos, 1999), the fusion algorithm (Santos Jr et al., 2009), and how probabilistic inferences are derived (Rosen et al., 2004). Then, we introduce a few new concepts such as *decomposability* and *substitutability* that allow us to derive a linear solution for probability computations.

3.3.1 Knowledge Representation

In this section, we first provide a formal definition of Bayesian Knowledge Bases (Santos Jr and Santos, 1999) which is a probablistic representation of knowledge. Then, we describe the fusion algorithm (Santos Jr et al., 2009) which allows us to merge multiple BKBs into one BKB without violating semantic and probablistic soundness.

Definition 7 (Correlation Graph)**.** A *correlation graph* $G = (I \cup S, E)$ is a directed graph such that I and S are disjoint, and $E \subseteq \{I \times S\} \cup \{S \times I\}$. Furthermore, for all $a \in S$, (a, b) and (a, b') are in E if and only if $b = b'$. $\{I \cup S\}$ are the *nodes* of G and E are the *edges* of G. A node in I is called an *instantiation node* (I-node) and a node in S is called a *support-node* (S-node).

I-nodes represent the various instantiations of ramdom variables (r.v.s), that is, an *assignment* of a value to a random variable. S-nodes, on the other hand, explicitly embody the relationships (conditional (in)dependence) between the I-nodes.

Notation. Let a be any node in $I \cup S$. $head_G(a) = \{b|(b, a) \in E\}$ are the *immediate predecessors* of a in graph G. $tail_G(a) = \{b|(a, b) \in E\}$ are the *immediate descendants* of a in graph G. In addition, we denote the r.v. of an I-node a by $rv(a)$, and its value assignment by $assn(a)$.

Let π be a partition on I. Each cell in π will denote the set of I-nodes (instantiations) which belong to a single r.v. and are mutually exclusive instantiations. For a variable X, we denote the set of all its possible instantiations by $D(X)$ - this set corresponds to the partition cell for variable X. In BKBs, we can represent random variables with discrete but multiple instantiations. We use RV to denote the collection of all r.v.s.

Definition 8. G is said to respect π if

1. for any S-node $b \in S$, the predeccessor I-nodes of b, $head_G(b)$, assigns at most one instantiation to each r.v.

2. for any two distinct S-nodes b_1 and b_2 in S such that $tail_G(b_1) = tail_G(b_2)$, there exists an I-node in $head_G(b_1)$ whose r.v. instantiations contradicts an I-node in $head_G(b_2)$. Furthermore, b_1 and b_2 are said to be mutually exclusive.

The above two conditions guarantees that the conditional (in)dependencies are meaningful especially during reasoning as we shall see later. The first condition simply prevents conditionals of the form $P(A = a| \ldots, B = b, \ldots, B = b', \ldots)$ where $b \neq b'$. For the second condition, consider some given head (or consequent) and the tals (or antecedent) of any two distinct S-nodes that share this head. If the two tails are *not* mutually exclusive (also called *compatible*), then they should be combined into a single more detailed condition. Condition two guarantees that no compatible S-nodes occur.

Definition 9 (Bayesian Knowledge Base)**.** A *Bayesian knowledge base (BKB) K* is a 3-tuple (G, w, π) where $G = (I \cup S, E)$ is a correlation graph, and w is a function from S to $[0, 1]$, π is a partition on I, and G respects π. Furthermore, for each $a \in S$, $w(a)$ is the *weight* of a.

Notation. We denote a cell of π by $\pi(i)$ and we denote the number of I-nodes contained in each cell by $|\pi(i)|$.

Definition 10 (BKB Fusion)**.** BKB Fusion (Santos Jr et al., 2009) is a process to *fuse* a collection of source BKBs into one BKB. Let $K' = (G', w', \pi')$ denote the target BKB to fuse from a collection of source BKBs $\{K_{i,i\in\{1,\ldots,m\}} = (G_i, w_i, \pi_i)\}$ where $G_i = (I_i \cup S_i)$.

K' is built by iterating through all the S-nodes in each source BKB. Basically, for each S-node, add a new I-node pointing to this S-node. The purpose of this new I-node is to denote the source of this S-node, thus, the name of the new I-node denotes the name of S-node's tail while the state of this I-node is the label of the source BKB. As we need to make sure the new graph G' respects π', we add an additional S-node supporting this new I-node. Intuitively, all the new S-nodes are *prior* nodes with no immediate predecessors. The immediate predecessors and immediate decendents, connected by a S-node, is a subgraph of source BKB. After adding the new I-node and the new S-node, simply merge this subgraph with K'. We have these relationship between each source BKB with the fused BKB:

- $\forall K_i, I_i \subseteq I'$
- $\forall K_i, S_i \subseteq S'$

Notation. Let $I'_s \subseteq I'$ denote the set of new I-nodes, $S'_s \subseteq S'$ denote the set of new S-nodes, and $RV'_s \subseteq RV'$ denote the set of new r.v.s introduced in the process of fusion. We have $G' = (I' \cup S', E')$ where $I' = \cup_{i=1}^{m}\{I_i\} \cup I'_s$ and $S' = \cup_{i=1}^{m}\{S_i\} \cup S'_s$.

K' has the following properties:

1. $|I'_s| = \sum_{i=1}^{m}(|RV_i|)$. The number of new I-nodes is the sum of the number of r.v.s. in each source BKB.

2. $|I'_s| \leq \sum_{i=1}^{m}(|\{S_i\}|)$. The number of new I-nodes is at most the same size as the number of S-nodes in all source BKBs.

3. $|S'_s| = |I'_s|$. The number of new S-nodes is the same as the number of new I-nodes.

4. $|RV'_s| = |\cup\{RV_i\}|$. The number of new r.v.s is the size of unions of r.v.s sets from all source BKBs.

The weight function w' for any S-node b in the new BKB K' is as the follows:

$$w'(b) = \begin{cases} w''(b) & \text{if } b \in S'_s \\ w_i(b) & \text{if } b \in S_i \end{cases}$$

All the weights of S-nodes from the original source BKBs are preserved. All the new S-nodes are supporting prior I-nodes. Function w'' assigns a weight to a *prior* I-node which is a new source node added in the process of fusion. As multiple I-nodes may be instantiations of the same r.v., the total sum of the weights should not exceed 1.0. One example of such a weighting function $w''(b)$ is to assign the S-node to each instantiation the same weight. Alternatively, the weight function can also take additional reliability information of each source BKB, and gets normalized so that the weights of all sources for a given random variable cannot exceed 1.0.

3.3.2 Probabilistic Inferencing

In probablistic reasoning, Bayesian Knowledge Bases can determine the probabilities of the following form $P(A_1 = a_1, \ldots, A_m = a_m | B_1 = b_1, \ldots, B_n = b_n)$ where the r.v.s $A_1, \ldots, A_m$ are our target r.v.s and $B_1 = b_1, \ldots, B_n = b_n$ are our evidence. Using these probabilities, we can answer questions such as:

1. What is the most probable state of the world given the evidence?

2. What is the most likely state of a r.v. given the evidence?

3. What is the most probable composite state of a set of r.v.s given the evidence?

In this section, we first describe *inference graphs* that is the key to computing probablities for the above questions. Then, we introduce some new definitions on the graphical relationship between source fragments with the fused fragment. Third, we describe how we compute probabilities.

BKB is a graphical structure of conditional (in)dependencies. The central idea is to consider relevant subgraphs of the BKB called inference graphs. Intuitively, these inference graphs represent partial instantiations to the world through their I-nodes while the product of the probabilities on their S-nodes will denote the joint probabilities of the partial instatiation.

Let $r = (I' \cup S', E')$ be a subgraph of our correlation-graph $G = (I \cup S, E)$ where $I' \subseteq I$, $S' \subseteq S$, and $E' \subseteq E$. Then, r has a weight $w(r)$ defined as follows: $w(r) = \prod_{s \in S'} w(s)$.

Definition 11. An I-node $a \in I'$ is said to be well-supported in r if there exists an edge (b, a) in E'. Furthermore, r is said to be well-supported if for all I-nodes a in I', a is well-supported.

Each I-node must have an incoming S-node in r.

Definition 12. An S-node $b \in S'$ is said to be well-founded in r if for all $(a, b) \in E$, $(a, b) \in E'$. Furthermore, r is said to be well-founded if for all S-nodes b in S', b is well-founded.

If an S-node b is present in r, then all incoming I-nodes (conditions) to b in G must also be present in r.

Definition 13. An S-node $b \in S'$ is said to be well-defined in r if there exists an edge $(b, a) \in E'$. Furthermore, r is said to be well-defined if for all S-nodes b in S', b is well-defined.

Each S-node in r must support some I-node in r.

Definition 14 (Inference Graph)**.** r is said to be an inference over K if r is well-supported, well-founded, well-defined, acyclic, and for all cells σ in π, $|I' \cap \sigma| \leq 1$, i.e., each r.v. has at most one instantiation in r. Furthermore, r is said to be a *complete inference* over K if for all cells σ in π, $|I' \cap \sigma| = 1$, i.e., each r.v. has one unique instantiation in r. r is said to be maximal in K if no proper superset of r is an inference over K.

Product $w(r)$ is the joint probability of the r.v. instatiations contained in the I-nodes of r.

In the previous section, we introduced the concept of fusion that combines multiple BKBs into one BKB. Through fusion, the joint probability distribution from each source BKB are fused into one global joint probability distribution. At the same time, each source BKB is still preserved and retrievable from the fused BKB.

Carefully selecting the inferences we consider, we can answer the various queries we described earlier. For example, suppose we wish to determine the most probable state of the world when given evidence set U. We only consider r.v. states that are reachable or can be reached from the evidence. If there is no undirected path from an r.v. state to evidence in U, the evidence is independent of the r.v. state.

We construct $R(U)$ as follows: Inference r is in $R(U)$ if r is connected, r is maximal, and $U \subseteq span(r)$[5]. Our goal is to determine the inference r^* with highest probability over all inferences in $R(U)$.

$$P(r^*|U) = \max_{r \in R(U)} P(r|U) \tag{3.1}$$

Note that $P(r|U) = \frac{P(r,U)}{P(U)} = \frac{P(r)}{P(U)}$. Since $P(U)$ is a constant, we only need to determine the most probable instantiation-set r^* in $R(U)$ such that $r \subseteq r^*$.

$$P(r^*|U) = \max_{r \in R(U)} w(r) \tag{3.2}$$

The most probable instantiation-set contains the most probable state of the world. If we are interested in the state of a particular r.v. A given evidence set U. The question now becomes: What is the most likely state of the world for an instantiation of a of A that is consistent with U? Similarly, we can construct $R(U)$ as follows: Inference r is in $R(U)$ if r is connected and $a \in span(r)$, and r is consistent with U. We can answer it by translating this question to the one above. Both this question and the previous one are *belief revision* type queries.

There is another type of reasoning query called *belief updating*: What is the likelihood of state of a r.v.s X given evidence U? We can answer this question by computing through belief revision. As BKB accommodates incompleteness in the knowledge, the exact probability is not available due to incomplete knowledge. However, we can obtain a lower bound and upper bound of the probablities for instantiations of r.v. X. The lower bound of $P(X|U)$ can be derived by summing up the joint probablities for all the inferences r that are in $R(U)$. To compute the

[5]We define the span of a set of I-nodes or an inference to be all the r.v.s instatiations involved in that set of inference.

upper bound of $P(X|U)$, we further construct a set of inferences $R'(U)$ from $R(U)$ by removing all the descendants of instantiations of X in each inference from $R(U)$. The upper bound is derived by summing up the joint probabilities for all the inferences of instantiations of X in each inference from $R'(U)$. Formally, we have:

$$P(X|U) \geq \sum_{r \in R(U)} P(X|U) \tag{3.3}$$

$$P(X|U) \leq \sum_{r \in R'(U)} P(X|U) \tag{3.4}$$

Note that $P(X|U) = \frac{P(X,U)}{P(U)} = \frac{P(X)}{P(U)}$ and the approximations we derive by summing all relevant inferences are both smaller or equal to the real lower and upper bounds.

$$\sum_{r \in R(U)} P(X|U) \approx [\sum_{r \in R(U)} w(r), \sum_{r \in R(U)} w(r)] \tag{3.5}$$

where $[\sum_{r \in R(U)} w(r), \sum_{r \in R(U)} w(r)]$ denotes lower and upper bound of the approximation.

Now we discuss relations between a collection of BKBs and the BKB fused from them.

Definition 15 (Source Fragment-Compatible Subgraph)**.** Let $K' = (G', w', \pi')$ be a BKB fused from set $\{K_i = (G_i, w_i, \pi_i) : i \in \{1, \ldots, m\}\}$. A subgraph g of K' is *compatible* with a source fragment K_i if g is well-supported, well-founded, well-defined, $I_{K_i} \subset I_g$, and $(I_{K_i} \setminus I_g) \subseteq I'_s$ where I'_s is a set of source I-nodes.

A subgraph compatible with a source fragment would contain all its I-nodes plus the additional new source I-nodes inserted in the process of fusion.

The relation between source fragments and source fragment-compatible subgraphs is surjective. In details, we have:

1. Each g is compatible with multiple source fragments if they have the same set of I-nodes. These source fragments can have different S-nodes and weight functions.

2. For every source fragment K, there exists exact one compatible g.

3. The union of all subgraphs g is the same as fused BKB K'.

Notation. We use $compat(K', K)$ to denote the subgraph of K' compatible with the source fragment K.

Definition 16 (Decomposable Subgraph)**.** Let $K' = (G', w', \pi')$ be a BKB fused from set $\{K_i = (G_i, w_i, \pi_i) : i \in \{1, \ldots, m\}\}$. A subgraph $g = compat(K', K_{i,i\in\{1,\ldots,m\}})$ is *decomposable* if all maximal inferences in g are complete.

Notation. We use $decom(K')$ to denote the set of decomposable subgraphs in K'. We have $0 \leq |decom(K')| \leq m$ as the number of source fragments is m.

Definition 17 (Decomposable BKB)**.** Let $K' = (G', w', \pi')$ be a BKB fused from set $\{K_i = (G_i, w_i, \pi_i) : i \in \{1, \ldots, m\}\}$. A BKB K' is *decomposable* if all its source fragment-compatible subgraphs are decomposable. For a *decomposable* BKB, we can answer the questions by integrating partial results from reasoning the question over each g. We can answer the belief revision type queries by answering the question on each $g \in decom(K')$. Formally, we have [6]:

$$P(r^*, U) \approx \max_{g\in decom(K')} \left(\max_{r\in R_r^g(U)} w(r) \right) \tag{3.6}$$

As belief updating can be derived through belief revision, we can also answer the belief update type queries by answering the question on each $g \in decom(K')$. Formally, we have:

$$P(X|U) \approx \sum_{g\in decom(K')} \left(\sum_{r\in R_r^g(U)} w(r) \right) \tag{3.7}$$

[6]The probability value is an approximation, ref to Equation 3.2.

Chapter 4

Double Transition Model

This chapter formally introduces Double Transition Model to address the learning aspect of an opinion formation task. In this chapter, first we will provide an overview of our effort in Phase I with a focus on justification of a new cognitive model. Next, we provide formal definitions of a DTM in mathematical terms. Then, we describe each component of a DTM independently: 1) the memory transition graph in a DTM, 2) the query transition graph in a DTM, 3) the links formed within a DTM. As a DTM is a cognitive model, in Section 5, we also compare it against a well-adopted cognitive architecture SOAR (Lehman et al., 2006) to highlight the differences.

4.1 Overview

In Phase I, our objective is to address the learning aspect of an opinion formation task; here learning means the internalization of external influence. There are a variety of things researchers have confirmed that can be internalized. Thus this phase focuses on the design of a cognitive model capturing the internal mechanisms embodying internalization.

We realize that a new design of a cognitive model is needed mainly for the reason that the majority of existing cognitive architectures follow a "divide-and-conquer" paradigm - the cognitive processes are treated independently and the variety of knowledge reprsentations are modeled separately as well (See the memory structure of SOAR (Lehman et al., 1996, 2006) in Figure 4.1a as a reference.). There are two main weaknesses with regards to the "divide-and-conquer" design:

- Human engineering effort

  Models designed with the philosophy of "divide-and-conquer" often results in more components. The research approach then turns to focus on the design of each individual component (what each component does, what each component contains) and the design of how these components interact with each other. The components are often of hetegenous type, i.e., one component is knowledge, another component can be a cognitive process.

  The main weakness of this design is expensive human engineering effort in initializing each component, e.g., Cyc has been a 30-year effort in assembling a comprehensive ontology[1] of everyday common sense knowledge.

- Butterfly effect

  The second weakness of this design is that the errors/assumptions in one component can have a butterfly effect on the performance of the entire system.

  Daniel Kahneman (Daniel, 2011) and Amos Tversky (Kahneman and Tversky, 1984) introduced the notion of cognitive biases to describe the mistakes individuals constantly and consistently make. Their studies suggested that human cognitive activities may not be in accord with existing mathematical models (Kahneman and Tversky, 1972). Therefore, the more components are manually engineered, the more likely that some components may have too many implicit assumptions that are not compatible with Bounded Rationality.

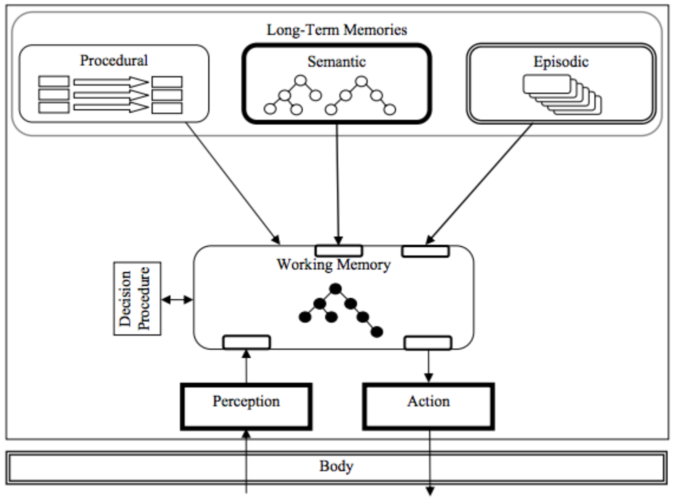
[1]The new release of OpenCyc has 239,000 terms, 2,093,000 triples downloadable from `http://www.cyc.com/platform/opencyc`.

Therefore we realize that a cognitive model with compact structure and minimal requirement of human engineering effort highly desired; that is, a model can be fully instantiated by observables from individuals without the needs for precoded domain knowledge. We are then triggered to propose the following idea: if we store the snapshots of all the states a cognitive model has experienced before, then the links between two connecting snapshots naturally capture the differences in knowledge and so forth. This was the origin idea that inspired us to design a Double Transition Model (see Figure 4.1b). For simplicity, we consider two forms of knowledge to build a DTM: learning episodes represent perception of the environment, and queries representing requests of opinions. A DTM is a *networked space of cognitive states, from each of which an opinion can be derived as a combination of the knowledge and the query within that state.*

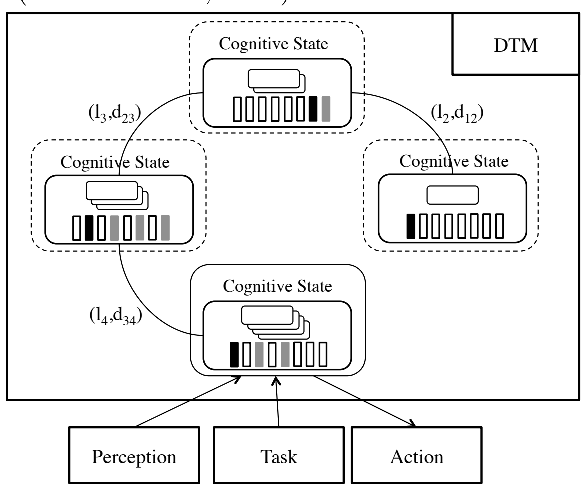
There are a couple of challenges in designing a DTM to embody the internalization of external influence. We summarize them here and the detailed approach to tackle them will be provided in the following sections. The key challenge for this phase is to make sure the space of cognitive states is sufficient to reflect diversities in human reasoning. This is extremely challenging particularly in the situation that the causes of diversified human reasoning have not reached an agreement yet by researchers. Thus the transformation rules between connecting states are driven by the desire to be flexible with the main theories of bounded rationality.

4.2 Definitions

A Double Transition Model (DTM) may be viewed as the cross product of two transition graphs — the query transition graph and the memory transition graph.



(a) Memory Structure in SOAR Cognitive Architecture from (Lehman et al., 2006)



(b) Double Transition Model

Figure 4.1: Comparison of SOAR and DTM

Definition 1 (Query Transition Graph)**.** A query transition graph Q is an undirected graph (V^Q, E^Q), where V^Q is a collection of partial instantiation of the random variables, such as $(A_1 = a_1, A_2 = a_2, A_3 = a_3)$... Each A_i is a random variable and the corresponding a_i is the value or instantiation associated with the A_i. There is an edge between two vertices v_1^Q and v_2^Q in V^Q if and only if v_1^Q can be transformed into v_2^Q using a single query transformation. It is clear from the definition of query transformation that if v_1^Q can be transformed into v_2^Q using a single query transformation, so can v_2^Q be transformed into v_1^Q using a single query transformation.

Definition 2 (Memory Transition Graph)**.** A memory transition graph K is an undirected graph (V^K, E^K), where V^K is a knowledge base (BKB in our case). There is an edge between two vertices v_1^K and v_2^K in V^K if and only if v_1^K can be transformed into v_2^K using a single domain knowledge transformation. It is clear from the definition of domain knowledge transformation that if v_1^K can be transformed into v_2^K using a single domain knowledge transformation, so can v_2^K be transformed into v_1^K using a single knowledge transformation.

Definition 3 (Double Transition Model)**.** Let $Q = (V^Q, E^Q)$ be a query transition graph and $K = (V^K, E^K)$ a domain knowledge transition graph. A DTM T induced by Q and K is the undirected graph (V^T, E^T) where $V^T = V^Q \times V^K$ and there is an edge between $v_1^T = (v_1^Q, v_1^K)$ and $v_2^T = (v_2^Q, v_2^K)$ in V^T if and only if 1) $v_1^Q = v_2^Q$ or $(v_1^Q, v_2^Q) \in E^Q$ and 2) $v_1^K = v_2^K$ or $(v_1^K, v_2^K) \in E^K$. In other word, T is the cross product of Q and K.

4.3 Memory Transition Graph

4.3.1 Motivation

An interesting observation from RAS model is that political survey responses are a function of the most immediately accessible considerations at the time when the survey respondent is trying to make a decision on his/her altitutde towards the survey question. People make decisions "on the top of their head" with the most salient consideration. People arrive at their attitudes by averaging across the relevant negative and positive considerations (Zaller, 1992).

These empirical findings implies the important role of knowledge in the opinion formation process and also implies there can be a strong relationship between the working-memory chosen to derive an opinion. This section focuses on building a memory transition graph that is compatible with main theories in bounded rationality. There is significant body of work describing three types of reasoning: inductive reasoning (Angluin and Smith, 1983), deductive reasoning (Rips, 1994), and abductive reasoning (Magnani, 2001). The discovery of three reasonings was inspired by observations of human reasoning. However, it has been consistently reported that human mix these types of reasoning to solve problems but the conditions for switching between different reasoning schemes are not clear (Arthur, 1994). Furturemore, even just conducting inductive reasoning, it has been consistently disocvered that human differ in the level of generality of knowledge to rely on (Hintikka, 1970).

In this thesis, we address the problem with regards to varying degree of generality in human reasoning. We cover two forms of inductive reasoning: episodic-based reasoning inferences over all features but is selective on learning episodes to include, and feature-based reasoning inferences over all learning episodes but is selective on features to consider. The episodic-based reasoning is consistent with theories on memory recall (Coleman and Mizel, 1968) and has also been found as one of the human heuristics to simplify a reasoning problem (Kahneman and Tversky, 1972; Nilsson et al., 2008; Schwenk, 2006). The needs to cover different levels of generality in inductive reasoning has been naturally addressed by the existing probabilistic inferencing, thus we would not cover it here. The capabilities for episodic-reasonings are fulfilled by the fushion methods (Santos Jr et al., 2009).

Another challenge we try to address is the incompleteness of perceptions of environment; it is natural that values of some features cannot be observed. It is important to capture incompleteness into the knowledge representation directly as it can also be later on used to trace the origins in different opinions. Furthermore, current methods that estimate missing values (Schafer and Graham, 2002; Troyanskaya et al., 2001) may introduce biases into the data for both whether this feature is truly observed and the reliabilities of the value under various contexts. To avoid adding biases into our framework, we choose Bayesian knowledge bases due to its nature in handing incomplete knowledge. Other methods can also be adopted to explore different possibilities, in this thesis we stick to BKB for all the system implementions and mathematical definitions.

The following section describes procedures to build a memory from perceived learning episodes, and methods to reduce computational complexity, in particular as follows:

1. Method to convert a learning episode (in the form of a vector) to a BKB fragment.
2. Method to convert a set of learning episodes (in the form of a matrix) to a BKB fragment.
3. A linear solution (algorithm) to compute joint probability of a query.

4.3.2 Knowledge Construction

Definition 4 (Feature Matrix)**.** A feature matrix, $\mathbf{L}$, stores all the *learning experiences*. The matrix is m *by* n where m is the number of learning episodes and n is the number of features. Each row $\mathbf{r}_{i,i\in 1,\ldots,m}$ of $\mathbf{L}$ is a *learning episode* and each column $\mathbf{c}_{j,j\in 1,\ldots,n}$ is a *feature vector*. Furthermore, we use $a[i,j]$ to denote a matrix cell and $a[i,j] \in \mathbb{C} \cup \{?\}$. The set $\mathbb{C}$ is a finite collection of values each cell can take, that $\mathbb{C} \subset \mathbb{Z}$ and $|\mathbb{C}| = C$. An example of a binary class can be $\mathbb{C} = \{0,1\}$. For convenience, we call $\mathbb{C}$ the *scope* of $\mathbf{L}$.

Notation. We label all features of $\mathbf{L}$ as "A_1",...,"A_n" and label all learning episodes as "B_1",...,"B_m" consistent with the subscripts in the feature matrix.

An episode of learning may involve one to many objects. One may learn by observing features of one object (called *single-object learning*) but may also learn by comparing features among multiple objects (e.g., *pair-wise learning* compares two objects). In the case of single-object learning, the value of a feature can refer to presense/absense of a property (e.g. A duck does <u>not</u> have tails.), discrete values from discretizing numerical-valued property (e.g. A duck has <u>two</u> feets.), or a qualitative description (e.g. A duck has <u>small</u> feets.). It is important to note that the value of a feature can be missing, which we denote its value by $\{?\}$.

Pair-wise learning can be obtained via *direct observation* or via combining two single-object learning episodes. Relations for a feature between two objects can be obtained directly via perception. For instance, one can learn that building A is taller than building B because he can see B's roof but cannot see A's. Therefore, one does not need to know the exact height of a building nor whether each building is tall or short to conduct a comparison. Alternatively, one can derive a pair-wise learning by mapping features for two single-object learning episodes to a multi-class set $\mathbb{C} \cup \{?\}$. An example of such a multi-class set is $\mathbb{C} = \{1, 2, 3\}$ where 1 refers to a large than relationship, 0 refers to a smaller than relationship, and 2 refers to equivalance. Here, a larger than relationship is derived from a literal comparison of feature values which may not equate with semantic meaning behind the feature values. Furthermore, if any of these two objects has a missing value for the same feature, that feature value of the pair-wise learning is considered be missing as well.

When there are numerical-valued features, a discretization procedure may be needed to transform a continuous space into discrete states. For instance, one may discretize a continuous space $[0, 40]$ to two discrete states 10 and 30. Values between $[0, 20)$ are mapped to 10 and values between $[20, 40]$ are mapped to 30. All values within the same interval are considered equal. Discretization is not needed if learning episodes are comparative learning episodes (e.g. pair-wise) thus comparative properties for numerical-valued features are preserved.

Another advantage of pair-wise learning episodes are the controllable size of class. If the target feature has a wide range of values, the inference task may become a multi-class classification task where the size of class can be large. However, the size of class can be as small as two (equal, else) or three (smaller, large, equal).

Similarly, we can classify questions into single-object question (Type I), pair-wise object question (Type II), and multi-object question (Type III). If a question asks a comparative relationship among two objects, it is straightforward to utilize pair-wise learning episodes. We can construct $\mathbf{L}'$ of pair-wise learning episodes from $\mathbf{L}$ of single-object learning episodes. On the other hand, if a question only focuses on one object, it is straightforward to utilize a $\mathbf{L}$ of single-object learning episodes but harder to utilize one of pair-wise learning episodes. In the later chapter, we will provide more in-depth discussion on the generality of questions and learning episodes. Basically, we propose an approach that can answer *single-object type questions*, *pair-wise object type questions*, and *multi-object type questions* based on an $\mathbf{L}$ of single-object learning objects without needing to constructing other matrices.

Definition 5 (Single-rooted Bayesian Knowledge Base)**.** Given a learning episode $\mathbf{r}$ with *scope* $\mathbb{C}$ in the form of a vector representation, our goal is to convert it into a BKB representation. As a BKB is a graphical representation, here we describe how to convert a vector into a BKB rooted at a target feature X.

We first construct a set of I-nodes I. For each cell $\mathbf{r}_{i\in\{1,\ldots,n\}}[i]$: If $\mathbf{r}_i \neq \{?\}$, we create one I-node a that $rv(a) = A_i$ and $assn(a) = \mathbf{r}_i$; if the value is missing, we create a set of I-nodes with size $\mathbb{C}$ where each I-node represents a possible instantiation of

$rv(a) = A_i$. After creating I-nodes for each cell, we add all these elements to set I. Let π be a partition over I.

Next step is to construct a correlation graph G. We construct a family of sets $\{I_i | span(I_i) = RV, I_i \subset I, |I_i| = n\}$. Intuitively, each subset of I contains exact one instiation of each r.v.. We call the root r.v. as X. For every subset I_i, we create a S-node connecting I-node $a \in I_i$ such that $rv(a) = X$ with all the other I-nodes in I_i. This S-node supports a and have other I-nodes as immediate predecessors. For every I-node $b \in I_i$ that $rv(b) \neq X$, we create a S-node supporting it. All the S-nodes along with the I-nodes connected by these S-nodes forms a subgraph. We build one subgraph for every I_i and merge it to graph G.

Notation. We use $K^X_{\mathbf{r}} = (G^X_{\mathbf{r}}, w^X_{\mathbf{r}}, \pi^X_{\mathbf{r}})$ to denote a BKB representation of a learning episode $\mathbf{r}$ rooted at r.v. X (can be called as a *feature-rooted BKB*).

We compute the weight of a node $b \in S$ such that $head_G(b) \neq \phi$ as follows:

$$w(b) = \begin{cases} \frac{1}{|\mathbb{C}|} & \text{if } assn(X) = \{?\} \\ 1.0 & \text{if } assn(desc_G(b)) = assn(X) \\ 0.0 & \text{if } assn(desc_G(b)) \neq assn(X) \end{cases}$$

The weight of a prior node $b \in S$ is computed as follows:

$$w(b) = \begin{cases} \frac{1}{|\mathbb{C}|} & \text{if } |\pi(head_G(b))| \neq 1 \\ 1.0 & \text{else} \end{cases}$$

Definition 6 (Non-Rooted Bayesian Knowledge Base)**.** The r.v. a BKB is rooted at denotes a dependency between its immediate predecessors and the rooted r.v. However, we may not know which r.v. is the target when elicitating the knowledge. When we do not have sufficient knowledge on the target r.v. or in general terms the dependency structure, one solution is to build a family of BKBs each rooting at a different r.v. Then we can fuse these single-rooted BKBs into a *non-rooted* BKB.

Notation. Let $K_{\mathbf{r}}^{X}$ denote a BKB of learning episode $\mathbf{r}$ rooted at feature X. Let $K_{\mathbf{r}} = (G_{\mathbf{r}}, w_{\mathbf{r}}, \pi_{\mathbf{r}})$ denote a non-rooted BKB of learning instance $\mathbf{r}$ by fusing source BKBs $K_{\mathbf{r}}^{A_j}$ where $j = 1, \ldots, N$ denoting each feature of the feature matrix.

Definition 7 (BKB Representation of Learning Experiences)**.** We construct a BKB $K(\mathbf{L})$ to cover all learning experiences by fusing all the BKB representation of each learning episode. This is achieved by using the fusion algorithm.

Notation. Let $K^{X}(\mathbf{L}) = (G^{X}(\mathbf{L}), w^{X}(\mathbf{L}), \pi^{X}(\mathbf{L}))$ denote a BKB of $\mathbf{L}$ fused from a collection of single-rooted BKBs $\{K_{\mathbf{r_i}}^{X} | i \in \{1, \ldots, m\}\}$.
Let $K(\mathbf{L}) = (G(\mathbf{L}), w(\mathbf{L}), \pi(\mathbf{L}))$ denote a BKB of $\mathbf{L}$ fused from a collection of non-rooted BKBs $\{K_{\mathbf{r_i}} | i \in \{1, \ldots, m\}\}$.

We use RV_A to denote r.v.s present in $\mathbf{L}$ and use RV_L to new r.v.s inserted in the process of BKB construction. We have $RV = RV_A \cup RV_L$ and $RV_A \cap RV_L = \phi$.

Theorem 8. Let $K^{X}(\mathbf{L})$ (abbrev. K^{X}) be a base knowledge from source fragments $\{K_i^{X} : 1 \leq i \leq m, m \in \mathbb{N}\}$. K^{X} is *decomposable.*

Proof. According to Definition 5, every maximal inference in a source fragment is complete. Formally, for every maximal inference r in K_i^{X}, $\mathit{span}(r) = |RV_{K_i}|$. For a subgraph g compatible with this source fragment, formally $g = \mathit{compat}(K^{X}, K_i)$, every maximal inference r' in g is also complete as $\mathit{span}(r') = 2|RV_{K_i}| = |RV_g|$.

Theorem 9 (Conditional Probability)**.** Now we discuss how to derive a closed-form solution for conditional probabilities. We can derive it via computing joint probabilities:

$$P_g(X = x_i | U) = \frac{P_g(X = x_i, U)}{\sum_{i=1}^{\mathbb{C}} P_g(X = x_i, U)} \tag{4.1}$$

As we are interested in the order of all states of X rather than the exact probablistic value, we can compare the joint probability rather than the conditional probability. Related proofs and the algorithm can be found in Appendix A and Appendix B.

4.4 Query Transition Graph

4.4.1 Motivation

In a query transition graph (See Figure 1.3), a query is a statement $(a_1, \ldots, a_n)$, which can be more explicitly written as $(A_1 = a_1, A_2 = a_2, \ldots, A_n = a_n)$ where $A_1, \ldots, A_n$ represent random variables (e.g. color) and $a_1, \ldots, a_n$ represent values (e.g. red) for the associated random variable. A query "$color = red, shape = round$" can be read as "color is red, and shape is round".

As we described as one of the key challenges/issues in Chapter 1, *Does the vector representation of a query suffice to represent human opinion requests?*

The answer is: unfortunately not. Natural language is our linguistic system for communication. Therefore, we need to explore first how realistic it is to represent a query in the form of vector representation.

The way humans initialize an opinion query is similar with how they initialize a search query for information seeking. In the process of information seeking, one raises the needs to seek additional information (called information needs). To describe his information needs to some one else, he further translates his thoughts into a communicatable form (called information request). The information request is usually represented through natural language but it may be further translated into key words or meta-info (called a search query) so that information retrieval systems can process (Baeza-Yates et al., 1999).

In the process of initializing an opinion query, one either receives an *opinion request* from someone else or forms one by himself from his initial opinion needs. In order to reason on it, one further translates an opinion request into a set of *opinion queries* that are of less ambiguity.

There may be many ways and situations under which opinion queries can be formed. In this thesis, we focus on forming opinion queries from an opinion request expressed through natural language. Now, we focus on addressing two important aspects in the rest of this section: 1) It is feasible to handle opinion requests that are in the form of natural language. We would like to demonstrate that even though a request can be in various linguistic forms, it is feasible to translate one form to another. 2) We can formalize an opinion task as a choice problem.

To ask for one's opinion, an opinion request must be in a form of a question. A question is an expression to make a request for information. Many categorization schemes exist regarding types of questions but we approach the problem using the following linguistic categorization:

- Yes-no question

  In linguistics, a yes-no question, formally known as a polar question, is a question whose expected answer is "yes" or "no". Formally, they present an exclusive disjunction, a pair of alternatives of which only one is acceptable. An example question is: Is today thanksgiving?

- Alternative question

  An alternative question (Karttunen, 1977) is formally known as a close-ended question, is a question whose expected answer is one from a given set of choices. For instance: “Is this building high or short?”

- Wh-questions

  Wh-questions use interrogative words to request information. Wh-questions is a type of open-ended questions, where the scope of the answers are not provided.

- Tag questions

  Tag questions are a grammatical structure in which a declarative statement or an imperative is turned into a question by adding an interrogative fragment, such as “right”. For example: Today is thanksgiving, is not it? A tag question can also be answered by a “yes” or “no”. The difference between a tag question and a yes-no question lies in the intent behind the question. A person raising a tag question aims to request an information to confirm his own answer. However, a person raising a yes-no question is mostly likely requesting an answer. Since the grammatical structure of tag questions can be converted to yes-no questions and there is no need to differentiate the intent behind a question, we consider tag questions being subsumed by the type of yes-no questions in this thesis.

An opinion request can be either expressed by a close-ended or open-ended question (Schuman and Presser, 1979). In the case of a close-ended opinion request, the scope of the answers is provided/implied by the request itself. We treat the task of answering an opinion request as a *choice* problem; that is, to select a value from ones included in the scope. Regarding alternative questions, the scope of answers are directly provided in the request. For instance, the scope of answer is “tall” or “short” for a request: Is this building tall or short? In the case of a yes-no question, the scope of answers are “yes” and “no”. When it comes to an open-ended question, even though the question does not imply the answers in a scope, however the opinion provider often has a scope in his own mind. For example, an open-ended opinion request could be: Who is the current president of United States? The opinion provider has a list of candidates that he can evaluate one by one.

Yes-no questoins, Wh-questions, and Tag questions can be translated to alternative questions. A possible translation solution between different types of questions are illustrated in Figure 4.2.

For instance, a yes-no question "Is this building tall?" is translated to an alternative question "Is this building tall or short?". A tag question , e.g. "This building is tall, is not it?", can first be translated into a yes-no question, e.g. "Is this building tall?", by modifying its grammar structure and then be further translated into an alternative question, e.g. "Is this building tall or short?".

When a yes-no question gets translated into an alternative question, we need to specify the full membership of alternative answers. However, it is worth paying attention to the scope of answers as they may not match between an opinion provider and an opinion requestor. For example, regarding a yes-no question: "Is Barack Obama the current president of United States?": the opinion requestor may be only uncertain whether the current president is Obama or Romney, but the opinion provider may consider all the past and current presidents of United States but not considering the presidential candidates. When we model an opinion provider's opinion formation process, we consider the scope of answers in the opinion provider's mind rather then that of the opinion requestor's mind. Taking differences in scopes of answers between an opinion provider and an opinion requestor is considered one important future work.

A Wh-question can be directly translated to an alternative-type question by

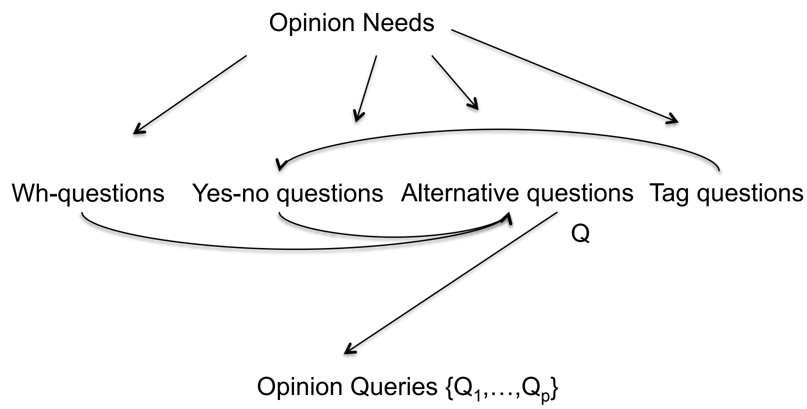


Figure 4.2: A Conversion Solution among Four Types of Questions

setting alternatives to be the candidate answers in an opinion provider's mind. It is important to note that not every Wh-question can be translated. For example, a question "What does a cloud look like?" is too abstract to be mapped to a scope of answers.

4.4.2 Query Construction

In this section, we address the knowledge representation of an opinion request, that of an opinion query, and translation between an opinion request to opinion queries. An opinion request is closely related to a question linguistically described by human. In last section, we discussed four different types of questions: Yes-no questions, Alternative questions, Wh-questions, and Tag questions. In general, a question is a linguistic expression to make a request for information, we define an opinion request as a question that makes a request for opinions.

In order to be able to answer an opinion request using our proposed framework, we discussed the feasibility and the scope of questions we can cover in the framework. We pointed out that: 1) it is feasible to cover both close-ended (yes-no, tag questions, alternative-type questions) and open-ended questions (Wh-questions), and 2) Yes-no questions, Wh-questions, and Tag Questions can be transformed to the alternative-type quesitons if sufficient information has been supplied.

As such, we design the knowledge representation of an opinion request in the form of alternative-type questions. We represent an opinion request using propositional logic where the description of the question is represented by the conjunction $\varphi_1^{o_1} \wedge \varphi_2^{o_2} \wedge \ldots \wedge \varphi_p^{o_k}$ and alternative candiate answers are represented by a disjunction $\zeta_{o_1}^{?} \vee \ldots \vee \zeta_{o_k}^{?}$, formally:

$$\varphi_1^{o_1} \wedge \varphi_2^{o_2} \wedge \ldots \wedge \varphi_p^{o_k} \models \zeta_{o_1}^{?} \vee \ldots \vee \zeta_{o_k}^{?}$$

where a φ represents a premise, the subscript denotes the index and the superscript denotes the object this premise refers to. A ζ represents a conclusion, the subscript denotes the object this conclusion refers to and the symbol ? is the probability attached to the conclusion given all the premises. Considering an opinion request "Is this building tall or short?" with the observation on its age, the propositional logic representation of this request looks like this:

$$\text{(age of the building=old)} \models \text{(height=tall)} \vee \text{(height=short)}$$

with one premise being:

$$\varphi^{o_1} = \text{(age of the building=old)}$$

and two conclusions each representing a candidate answer (a.k.a an opinion):

$$1^{st} \text{ conclusion} : \zeta_{o_1} = \text{(height=tall)}$$

$$2^{nd} \text{ conclusion} : \zeta_{o_1} = \text{(height=short)}$$

We can interprete the propositional logic representation of this opinion request as: By knowing that the age of this building is old, how to make a conclusion that a building is tall or short? Whether the entailment holds true or not is evaluated by assessing the probability of each conclusion; that is, the probability of the building being tall considering it is an old building, and the probability of the building short considering it is an old building. As the conclusions each describe an alternative answer, therefore we require that the conjunctive normal form of each conclusion is different.

In the premise, "age of the building=old" is a positional variable where "age of the building" is a propositional string while "old" is an interpretation of this propositional string. Propositional logic allows us to represent a complex logic using logic connectors (e.g. $\vee, \wedge, \neg$). For example:

$$\varphi^{o_1} = [((\text{type of the building=business}) \wedge (\text{age of the building=old}))$$

$$\vee ((\text{type of the building=government}) \wedge (\text{age of the building=young}))]$$

In this premise, multiple aspects are connected by the disjunctive operator $\vee$ while the multiple features expected to satisfy together are connected by the conjunctive operator $\wedge$. Similarly, a conclusion can be described by complex logic.

Now, within the premise, what is the vocabulary of propositional variables? It is intuitive to relate a propositional variable to a feature, but how about an object? As a question may be concerned with a relationship between multiple objects, it is beneficial if the model can handle relational questions. For instance, the question may look like: "Which building is taller, building A or building B?" Furthermore, more information may be supplied describing characteristics of each building to facilitate answering this question. In designing the syntax of the propositional logic expression of an opinion request, we include entity of a premise or a conclusion to denote the object it refers to.

Now we proceed with the representation of an opinion query. As an opinion query guides opinion reasoning, thus its representation is closely related to the knowledge representation of domain knowledge and inference mechanisms. In this thesis, Bayesian Knowledge Bases is our choice of knowledge representation since it models incomplete knowledge, with inference capabilities to answer different forms of probablistic questions.

We now provide formal definitions of an opinion request, an opinion query, and most importantly how an opinion request maps to opinion queries. We begin with the terms of propositional logic we use in this thesis to avoid confusion on terminology usage.

Terms of Propositional Logic:

Propositional variables:

$\{A_1, \ldots, A_n\}$ where A_i denotes a feature in our opinion model.

Multi-valued Interpretation:

Interpretation of a propositional variable is a value assignment to the variable. For instance, "height=tall" is a proposition where "height" is a propositional variable represented by a string and "tall" is an interpretation of variable "height". A propositional variable may have multiple interpretations.

Proposition:

A propositional variable along with an interpretation forms an atomic proposition.

Logical Operators:

A logical operator is a symbol to connect two or more propositions. Some common logical connectives include: $\vee, \wedge, \neg, \implies, \models$ etc. A proposition can be considered as either atomic propositions, and composite propositions that are composed by recursively applying logical operators to propositions.

Premise and Conclusion:

A premise and a conclusion is a proposition plus an additional entity mark. In the case of a premise, the object the premise refers to is denoted by a subscript. In the case of a conclusion, the object the conclusion refers to is denoted by a superscript. The entity mark facilitates the process to map an opinion request to a set of opinion queries and the latter can be used to select one from all candidate answers. However, it does not change the propositional language itself.

Definition 10 (Representation of an Opinion Request).

An **Opinion Request** R is formalized as:

$$\varphi_1^{o_1} \wedge \varphi_2^{o_2} \wedge \ldots \wedge \varphi_p^{o_k} \models \zeta_{o_1}^{?} \vee \ldots \vee \zeta_{o_k}^{?} \tag{4.2}$$

where $\varphi_1^{o_1} \wedge \varphi_2^{o_2} \wedge \ldots \wedge \varphi_p^{o_k}$ are premises and $\zeta_{o_1}^{?} \vee \ldots \vee \zeta_{o_k}^{?}$ are conclusions drawn from premises. p denotes the total number of premises and k denotes the total number of alternatives. In addition, we have:

For an object, say o_1, $0 \leq |\varphi_j^{o_1}| \leq p$ such that j is an index of a premise and $j \in \mathbb{N}$.

For an object, say o_1, $0 \leq |\zeta^{o_1}| \leq k$.

Intuitively, for each object, we can have zero to many premises and conclusions about it.

Definition 11 (Representation of an Opinion Query).

An **Opinion Query** is formalized as:

$$Q = (A_1 = a_1, \ldots, A_n = a_n) \tag{4.3}$$

where $n \in \mathbb{N}$. A_j that $j \in \mathbb{N}$ denotes a r.v. in a BKB which corresponds to a propositional varialbe in an opinion request. $A_j = a$ denotes a value assignment a to r.v. A_j that corresponds to an atomic proposition in an opinion request. For any two r.v.s, say A_i and A_j that $i, j \in \mathbb{N}$, we have $i \neq j$.

We call the set of instantiations of various r.v.s specified in an opinion query as a *state* of a world. The span of the world is defined as the number of r.v.s included in an opinion query. We have $1 \leq |span(state)| \leq n$.

Definition 12 (Translation of an Opinion Request)**.**

We define function $\Gamma(R) = \{Q_k : k \in \mathbb{N}\}$ as a translation function from an opinion request to a set of queries.

Given an opinion request R, the function $\Gamma(R)$ generates a list of queries:

Step 1: Partition all premises and conclusions into partition sets π_1 (for premises) and π_2 (for conclusions) according to it entity mark. As a result, all premises or conclusions in the same cell have the same entity marks.

Step 2: For each conclusion ζ_{o_i} that $i \in \mathbb{N}$ in a cell $\sigma_1 \in \pi_2$, find the cell in $\sigma_2 \in \pi_1$ with the same entity marks. Then, connect all the premises with same entity mark by conjunctive operators, denoted by τ_{o_i}.

Step 3: For every pair $(\zeta_{o_i}, \tau_{o_i})$ with the same entity mark, we form a set of opinion queries. First we translate both ζ_{o_i} and τ_{o_i} into disjunctive normal form. Then, for every conjunctive in a disjunctive normal form of ζ_{o_i}, we form one query for every conjunctive in τ_{o_i}. To make it clearer, if we have m conjunctive terms in the disjunctive normal form of ζ_{o_i}, m queries will be formed.

4.5 Transformations

In this section, we define *rules* to transform an opinion query and a working memory. In brief, we define rules of transforming a query in order to select worlds and define rules of transforming a working memory in order to rewrite the content in these worlds.

To illustrate our idea, let us take a look at a simple knowledge representation in the form of a matrix (Figure 4.3a). Say we are concerned with feature A_5 that may have a value 0 or 1, we can imagine two sets of worlds. Within one set of worlds, the feature A_5 in every possible world has value 0. In the other set of worlds, the feature A_5 in every possible world has value 1. In Figure 4.3a, we have $A_5 = 0$ in the set of worlds $\{w_3\}$ and $A_5 = 1$ in the set of worlds $\{w_1, w_2\}$.

Now in a new world, an individual cannot observe the value for feature A_5 for some reason. He may try to infer a likely value for A_5 by examining the values of this featue in the worlds he had been in in the past. It may be safe to claim that the value for A_5 is 1 simply because he has seen 1 twice but only seen 1 once in the past. Perhaps he has observed values for others feature that are relevant to feature A_5, he may selectively look at the previous worlds in which observations match. Let us say he has observed that $A_1 = 0$ and $A_3 = 1$ which are relevant to feature A_5 of interests. By just relying on observation that $A_1 = 0$, he would be undecided (situation in Figure 4.3b, the worlds that match are circled by red-dotted rectangles). When he relies on both observations $A_1 = 0$ and $A_3 = 1$, he is likely to conclude that $A_5 = 1$ (situation in Figure 4.3c).

The worlds can also be rewritten; that is, the worlds can be refactored. Some examples of rewriting are shown in Figure 4.3d-4.3f. Worlds of less importance may be removed to allow new space to store more important worlds. Similarly, features

	w_1	w_2	w_3
A_1	0	1	0
A_2	1	0	1
A_3	1	1	0
A_4	0	1	1
A_5	1	1	0

(a) Matrix of Three Worlds

	w_1	w_2	w_3
A_1	0	1	0
A_2	1	0	1
A_3	1	1	0
A_4	0	1	1
A_5	1	1	0

(b) $A_1 = 0$

	w_1	w_2	w_3
A_1	0	1	0
A_2	1	0	1
A_3	1	1	0
A_4	0	1	1
A_5	1	1	0

(c) $A_3 = 1$, $A_1 = 0$

	w_2	w_3
A_1	1	0
A_2	0	1
A_3	1	0
A_4	1	1
A_5	1	0

(d) Remove World w_1

	w_1	w_2	w_3
A_3	1	1	0
A_4	0	1	1
A_5	1	1	0

(e) Remove Features A_1, A_2

	w_1	w_2	w_3
A_1	0	0	0
A_2	1	0	1
A_3	1	1	0
A_4	0	1	1
A_5	1	1	0

(f) Flip Feature Value

Figure 4.3: Feature-Rooted BKBs over Feature A_3 for Different Learning Episodes

that are rarely involved may be removed from the worlds as well. With different transformation rules of knowledge applied, the final worlds being looked at can be dramatically different. In Figure 4.3d, no world gets retrieved by retrieving worlds containing $A_1 = 0$, $A_3 = 1$ from the worlds with world w_1 removed. In Figure 4.3e, rows of feature A_1 are removed which relax the searching criteria for worlds containing $A_1 = 0$ and $A_3 = 1$.

Through transformation, the final knowledge determines *how* the worlds would look like (content) and the final query determines *what* worlds to look at. The final query and knowledge altogether determine the knowledge that will be relied upon for reasoning. In this section, we define rules of query transformation motivated by this idea of *selection*; that is, the query guides how stored memory gets retrieved. We define rules of knowledge transformation motivated by the idea of *rewriting*; that is, the rewriting guides how the working memory might look like.

4.5.1 Query Transformation

A query is in the form of $a_1, \ldots, a_n$ (as an abbreviation of $A_1 = a_1, \ldots, A_n = a_n$). We define two transformation rules on a query: insertion rule to insert a new instantiation into a query and deletion rule to remove an instantiation from a query.

Let I^K denotes the set of all possible instantiations available from a BKB. Let I^q denotes the set of all instantiations (states) in a query and we have: $I^K = I^q \cup (I^{\neg q})$.

Definition 13 (Rule of Insertion)**.** The rule of adding a new instantiation can be formalized as an operation of moving one element from set $I^{\neg q}$ to I^q.

Notation (Notation)**.** We denote this operation by $a(x)$ where x is an instantiation.

Definition 14 (Rule of Deletion)**.** The rule of removing a state can be formalized as operation of moving one element from set I^q to set $I^{\neg q}$.

Notation (Notation)**.** We denote this operation by $r(x)$ where x is an instantiation.

4.5.2 Memory Transformation

As we have illustrated at the beginning of this section, the roles of query transformations are to select worlds while the role of memory transformation are to rewrite these worlds. However, how to rewrite the worlds? What are the principles behind rewriting these worlds? Furthermore, what are the impacts of the rewriting rules to the BKB representation?

Before provide mathmatical descriptions of the memory transformation rules, we first point out a few problems with possible memory transformation rules. Some examples of *desirable* world rewritting are: removing worlds (Figure 4.3d), decreasing the resolution of worlds (Figure 4.3e), and creating new worlds due to value changes in old worlds (Figure 4.3f).

Worlds are called inference graphs in a BKB. A world can be represented as a chain of if-then rules in a BKB and its probability can be obtained by multiplying the weights of all if-then rules. As a BKB is natural to incompleteness implies that it is flexible with different operations on if-then rules (removal, rewiring, insertion) as long as a BKB still remains valid and does not violate mutual-exclusivity. Here we introduce one principle for knowledge engineering: Every world has full value assignments for all features. Two examples of transformation rules that violate this principle are shown in Figure 4.4a and 4.4b. When only half a world is removed (Figure 4.4a), it becomes ambiguous when the query attempts to select worlds. As the value for A_3 is now inaccessible in world w_1, should a query collect this world or go examine the next one? The same ambiguity exists when parts of a feature are removed from all worlds (Figure 4.4b). Thus, the principle has resulted in the following consequences of knowledge transformation rules:

1) Full elimination of one or more features result in lower *resolution* of the knowledge (since all values of a feature must be removed from all worlds).

2) Full elimination of one or more worlds result in lower *size* of the knowledge (since all values of a feature in that world must be removed).

3) No elimination of features and worlds but the weights of worlds are modified.

4) Other types of modifications on if-then rules can result in a hybrid of 1), 2), and 3).

We can define a set of memory transformation rules and requirements that guarentee the worlds would still have full value assignments after transformation. However, the rules will be hard to use for knowledge engineers to express how they want to transform memory. To provide full flexiblity for knowledge engineers

	w_1	w_2	w_3
A_1	0	1	0
A_2		0	1
A_3		1	0
A_4		1	1
A_5		1	0

(a) Remove World w_1

	w_1	w_2	w_3
A_1			0
A_2	1	0	1
A_3	1	1	0
A_4	0	1	1
A_5	1	1	0

(b) Remove Feature A_1

Figure 4.4: Undesirable Transformation Results

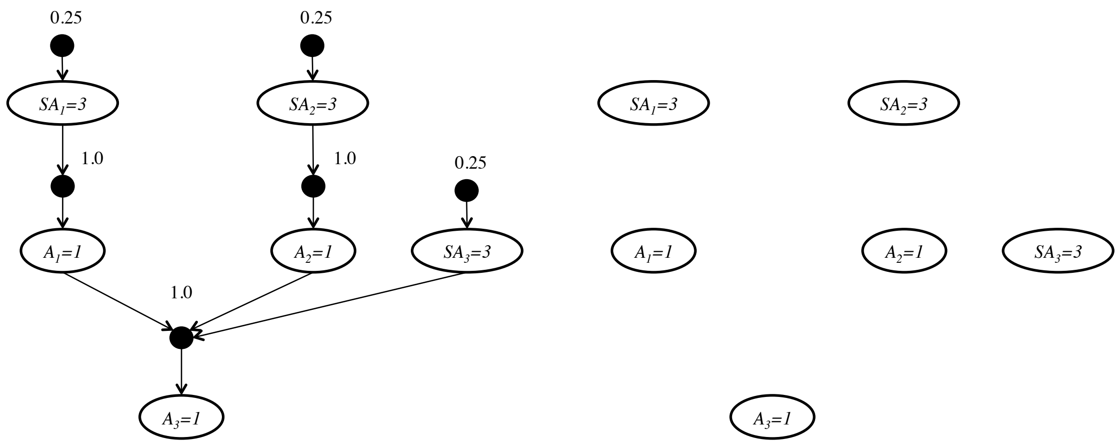
to define the knowledge transformation they want to study/model, we define the transformation rules in the form of graph substitution.

As a BKB is a graphical representation of a collection of if-then rules, the memory transformation can be defined as modifying these if-then rules. Thus, the substitution rule requires two inputs, one being the original if-then rules within the memory and the other being the new if-then rules to replace the old ones. Both if-then rules can be considered as a correlation graph of a BKB.

Definition 15 (Rule of Substitution)**.** Let K denotes memory, and pair $(K_1 \subseteq K, K_1')$ denotes one piece of auxiliary memory. We have $K = (G = (I \cup S), w, \pi)$, $K_1 = (G_1 = (I_1 \cup S_1), w_1, \pi_1)$, and $K_1' = (G_1' = (I_1' \cup S_1'), w_1', \pi_1')$. We require $I_1' = I_1$. In addition, we also require that a source I-node cannot support a different I-node. In details, an instantiation of r.v. SA_i such that $1 \leq i \leq n$ must point to a S-node linking to an instantiation of r.v. A_i.

The rule of substitution is an operation to replace all If-then rules captured in one BKB K_1 in K by all If-then rules defined in another BKB K_1'. In other words, K_1 marks the If-then rules to remove from K and K_1' defines the new If-then rules to insert into K.

Notation. We denote this operation by $subst(K, K')$ where K is a BKB containing If-then rules to be replaced by ones in K' that will result in a valid BKB with full value assignments after transformation.

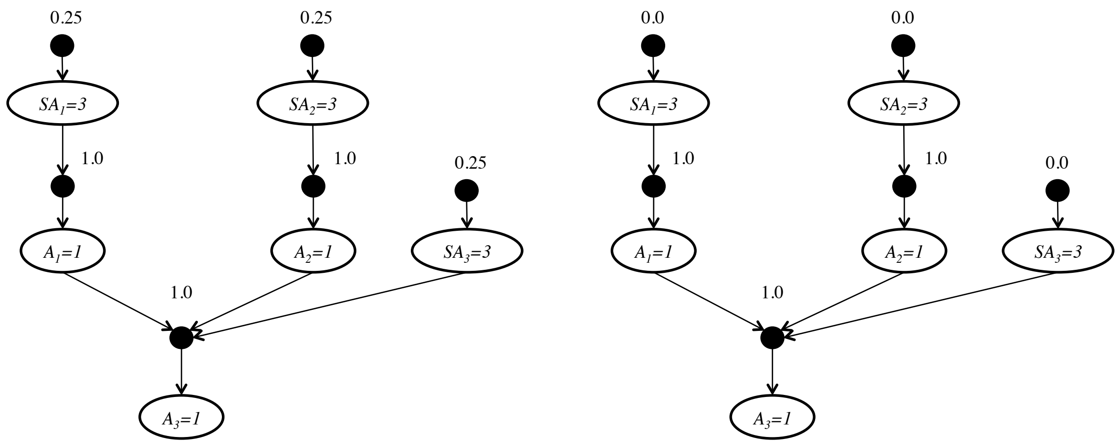


(a) K_1 (b) K_1'

Figure 4.5: Removal of One World

To illustrate how subsitution rule can rewrite the worlds, we demonstrate three examples of knowledge transformation. In Figure 4.5, we demonstrate a piece of auxiliary memory (K_1, K_1') that instructs removal of a world $(A_1 = 1, A_2 = 1, A_3 = 1, SA_1 = 3, SA_2 = 3, SA_3 = 3)$. An If-then rule is graphically represented as a S-node with its parents and children. An S-node may have no parents (For example, the S-node pointing to I-node $SA_1 = 3$.). K_1' has instructed that all the If-then rules that are in K_1 should be removed from memory K. According to K_1, K_1', we can now transformation memory K into K' by removing the marked If-then rules. The world $(A_1 = 1, A_2 = 1, A_3 = 1, SA_1 = 3, SA_2 = 3, SA_3 = 3)$ then gets removed from memory, the size of the memory is reduced by 1.

Figure 4.6 demonstrates the case of modifying the weight of a world $(A_1 = 1, A_2 = 1, A_3 = 1, SA_1 = 3, SA_2 = 3, SA_3 = 3)$. Both K_1 and K_1'' have been instructed to replace the If-then rules pointing to the prior I-nodes with the ones with probability 0. The weight of the world in the memory K can be calculated simply by multiplying all the weights of involved If-then rules: $P = 0.25 \times 0.25 \times 1.0 \times 1.0 \times 0.25 \times 1.0 = 0.015625$ and now it becomes $P'' = 0$.

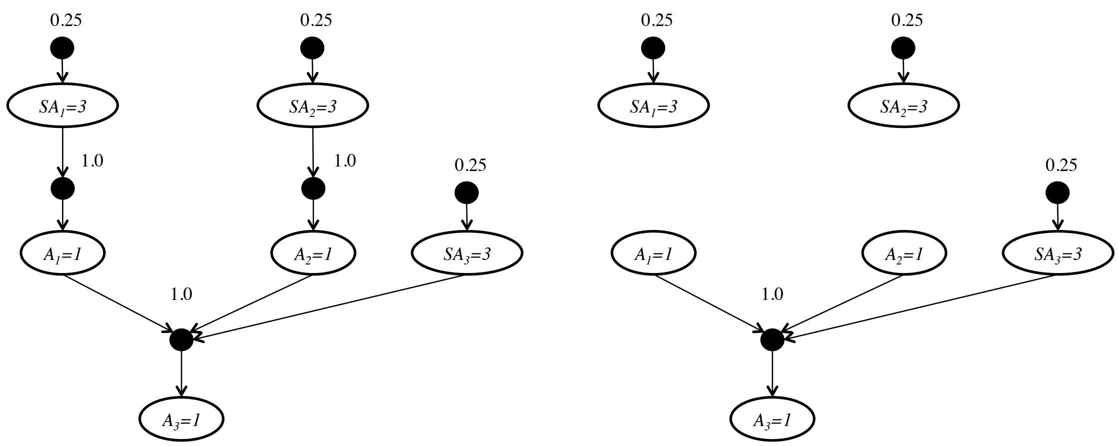


(a) K_1 (b) K_1''

Figure 4.6: Modify the Weight of a World

Figure 4.7 demonstrates the case of removing features SA_1 and SA_2. By removing If-then rules connecting features SA_1 and SA_2 with other I-nodes in a BKB, these two features are no longer reachable. Thus, the resolution of the memory is reduced by 2.

The 3^{rd} example may violate the requirement that a transformed memory needs to have full value assignments. For example, let us assume the memory looks like the one in Figure 5.5. After removing this If-then rule, the resolution of the world is reduced by two. However, the resolution of other worlds remain the same. The inconsistent resolution of worlds violates the requirement that all worlds have full value assignments. Therefore, we require that the substitution rule must result in a valid BKB with full value assignments. We can fix the problem presented in this example by explicitly including all I-nodes of feature SA_1 and SA_2 and removing all If-then rules involving them.



(a) K_1 (b) K_1'''

Figure 4.7: Remove Features

Chapter 5

Modeling Opinion Formation

In the previous chapter, we described the design of the DTM including the internal reasoning mechanisms, construction of the knowledge representation, and construction of queries, to the translation from an opinion request expressed by natural language to queries. In this chapter, we provide a walkthrough example on how a DTM computes an opinion. As the DTM only addresses the learning aspect of opinion formation tasks, here we demonstrate the examples motivated by interesting problems we observe in opinion diversity (motivated and described by a toy problem in the first section).

Then, we evaluate the design of a DTM by modeling four commonly-accepted human reasoning heuristics.

5.1 A Toy Problem

Consider a world with two rooms. Multiple agents walk around and can observe features of the objects in each room. The features of each object are partially observable to each agent. The agents cannot see the objects in the other room. From time to time, an agent in one room may ask the agents in the other room to give an "opinion" in one of these forms:

1. Type I: state of a single feature of an object. (e.g. Is object o_1 tall or short?)
2. Type II: relationship of a feature between two objects. (e.g. Is object o_1 taller than object o_2?)
3. Type III: relationship of a feature among multiple objects. (e.g. Among objects o_1, o_2, and o_3, which object is tallest?)

An example of such a world is shown in Figure 5.1. The tower room has three objects o_1, o_2, and o_3 and the bridge room has objects o_4, o_5, and o_6. Agents α and δ walk in the tower room, and agents β and γ walk in the bridge room. Every agent gathers learning episodes (shown in Table 5.1) by observing objects in its own room. In Table 5.1, each column represents a feature or property of an object and each row captures the observed values for all features for a particular object. The symbol ? in the table indicates a missing value.

Imagine that agent β wants to know more about features of objects in its room that it cannot perceive, it may ask agents α and δ who it considers to be capable of providing better opinions:

Q_1: What is the value of feature A_4 for o_4? *(Type I Question)*

Q_2: Compared to o_5, is the value of feature A_4 for o_4 larger or smaller? *(Type II Question)*

Q_3: What is the value of feature A_5 for o_4? *(Type I Question)*

Q_4: Compared to o_5, is the value of feature A_5 for o_4 larger or smaller? *(Type II Question)*

Q_5: What is the value of feature A_6 for o_4? *(Type I Question)*

Q_6: Among all the objects o_4, o_5, and o_6, which one has the highest value for feature A_6? *(Type III Question)*

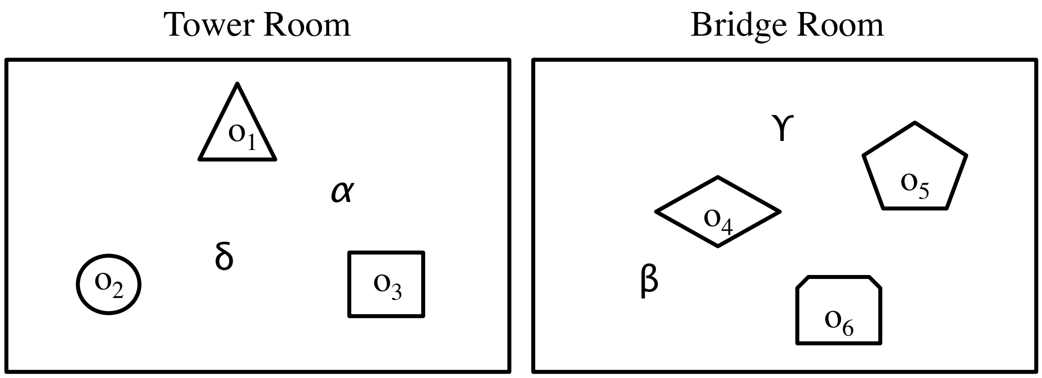


Figure 5.1: A World of the Toy Problem (Toy Problem 1)

Table 5.1: Learning Episodes Gathered by Agents (Toy Problem 1)

Room	Learning Episode	Object	A_1	A_2	A_3	A_4	A_5	A_6
Tower Room	r_1	o_1	1	0	1	2	?	1
	r_2	o_2	0	1	0	3	?	?
	r_3	o_3	0	0	0	1	?	?
Bridge Room	r_4	o_4	0	1	1	?	?	?
	r_5	o_5	1	0	1	?	?	1
	r_6	o_6	0	0	0	1	1	?

Although it seems simple, the toy problem captures some basic knowledge sharing phenomena within a society. Humans are in their own "sandboxes" – their occupations in a social hierarchy, their information channels and so forth to form their own learning experiences. One observes "objects" in his own sandbox and when one wants to know more, he either thinks on his own or asks others. If he needs to ask someone else, he can either ask other individuals in the same sandbox or asks other individuals in other sandboxes. Among all the sandboxes, some sandboxes have more objects relevant to his question - we call the individuals living in those sandboxes subject-matter experts. Intuitively, one would prefer getting opinions from a sandbox of subject-matter experts rather than from a randomly chosen sandbox. With the use of World Wide Web, we have more sandboxes available to choose from. As a consequence, the presence of the web has made experts reachable, identifiable, and searchable. But more importantly, one can easily obtain multiple opinions at once from the web (e.g. A student looks up ratings of a book provided by a large number of readers on Amazon.) which is a paradigm shift from how knowledge sharing occurs in the real world (e.g. A student asks for a librarian's opinion on a book.) crowd-sourcing.

This increasing use of the web has also brought attention to these two problems:

1. How to find the "right" individual of wisdom?

2. How to find the "right" sandbox of wisdom?

The significance of the first problem is self-evident: We do not have any good means to evaluate whether a group has provided a *wise* opinion. In the real-world, individuals both *learn* and *reason* differently which result in different opinions to deliver. For example, agent α only walks in the room during the day time and agent δ only comes out during the night. Naturally, agent δ may neglect a few objects due to poor lighting conditions. On the other hand, α may not be able to perceive features that are only observable during night time. For example, one feature can be the color of the neon light but the neon light only gets turned on at night. In terms of reasoning, everyone has his own choices of approaches in deriving an opinion. Agent α may neglect the color of a neon light as it is inexperienced with how this feature relates to the target question (e.g. Is this light expensive?). Agent β may only rely on part of its learning experience in order to speed up the process of delivering an opinion. If a sandbox is full of "quick-thinkers", β may wish to find another sandbox of "slow-thinkers" to seek more diverse opinions. It is hard to evaluate whether opinion differences are results of differences in learning and reasoning or results of errors or biases.

To illustrate the significance of the second problem, let us take a look at some complexities in answering questions Q_1 to Q_6 for the toy problem. To answer Q_1, the agents in the sandbox (tower room) are more likely to provide a better answer compared to the ones in the bridge room. The agents in the tower room are more *experienced* with regards to feature A_4 as they have three learning episodes with explicit observations. On the other hand, the agents in the bridge room have only one learning episode with explicit observation. This also explains why agent β attempts to ask the agents in the other room as its experience with feature A_4 is limited. Regarding Q_1, it is likely that β would benefit from the wisdom of the crowd in the tower room. However, the situation is not as clear for question Q_3 and Q_5. The agents in two different rooms have about the same experience on Q_5 (one learning episode with explicit values) and agents in the tower room are more experienced in answering Q_3. As β cannot see the objects in another sandbox, it may end up relying on a less robust opinion, and more ironically, β is already the most experienced one to answer Q_3.

Currently, the main research effort has been to investigate effective indicators of high-quality and low-quality opinions, however the common problem is its cross-domain incompatibilities. An indicator/method that works for one problem domain may no longer be as effective when migrated to another problem domain. In addition, we still have many long-standing questions and controversies on the topic of human reasoning and learning that are unresolved. This thesis is motivated towards reaching a better understanding on the process of opinion formation and to come up with a means to better tackle these two challenging problems.

5.2 Examples

5.2.1 Opinion Formation Process

We use a sample toy problem described below.

Task Specification

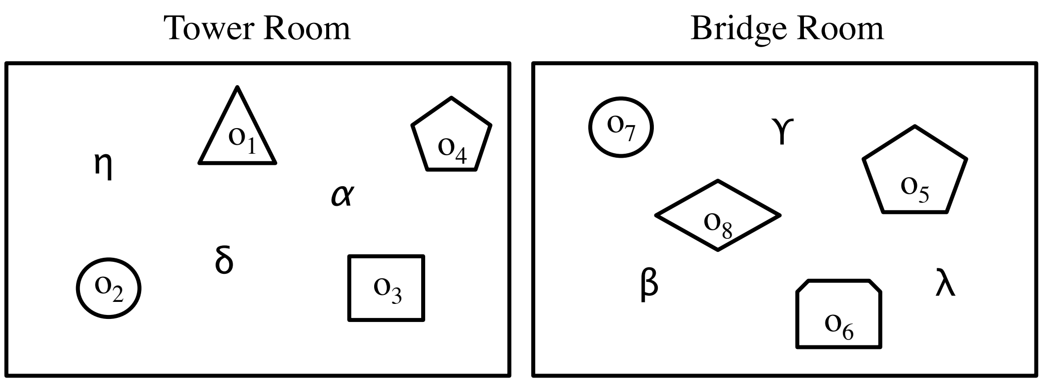


Figure 5.2: A World of the Toy Problem (Toy Problem 2)

Table 5.2: Learning Episodes Gathered by Agents (Toy Problem 2)

Room	Learning Episode	A_1	A_2	A_3
Tower Room	o_1	?	1	0
	o_2	0	1	?
	o_3	1	1	1
	o_4	?	?	?
Bridge Room	o_5	0	1	?
	o_6	1	0	?
	o_7	?	0	?
	o_8	0	1	?

The world has two sandboxes: tower room and bridge room. Each room has four objects and three agents as shown in Fig. 5.2. We have four explicit rules for the world:

1. All the agents in the same room acquire the same learning experience.

2. The agents can only see the objects in their own room.

3. Features are only partially observable.
4. All observations are binary valued.

Table 5.2 shows the learning experience of each agent. A_1, A_2, A_3 are features observed (e.g. whether an object is tall or short) by each agent. Rows 1-4 are learning experiences for agent α, δ and η, rows 5-8 are learning experiences for agent β, γ, and λ (rule #1 and #2). All the agents obtained partial observations for features $A_1 - A_3$(rule #3).

Now agent β is interested in knowing more about the missing value, thus he poses the following question:

Question: "What is value for feature A_3 of object o_5?"

As our focus is on modeling the behavior behind opinion formation, we assume that β decides to obtain opinions from the agents (α, δ, and η) in the tower room. In the following sections, we will illustrate how our framework models the opinion formation process of agents α, δ, and η. To illustrate the idea, we first demonstrate how a *non-regret* opinion is formed; that is, neither the working memory nor the queries are modified.

Forming an Opinion

We begin with the opinion request posed by agent β: "What is value for feature A_3 of object o_5 with observations that $A_1 = 0$ and $A_2 = 1$?" This is an opinion request expressed in the form of a Wh-question. To model the opinion formation process, we first translate this question expressed by natural langauge to an opinion request expressed by propositional logic. An example of a possible translation looks like this:

Opinion Request: $(A_1 = 0)_1^{o_5} \wedge (A_2 = 1)_2^{o_5} \models (A_3 = 0)_{o_5}^{?} \vee (A_3 = 1)_{o_5}^{?}$.

This opinion request can be read as: "Given observations on two features $(A_1 = 0, A_2 = 1)$ of object o_5, is the value of feature A_3 0 or 1?" Now, the opinion model translates this opinion request to a set of opinion queries following the procedure described in Definition 12 in Chapter 4. The opinion queries generated look like the following:

Opinion Queries: $Q_1 = (A_1 = 0, A_2 = 1, A_3 = 0)$, $Q_2 = (A_1 = 0, A_2 = 1, A_3 = 1)$

Each opinion query specifies the state of a world for each candidate answer. Now the DTM can reason the probability of each world via conducting belief updates over memory. To illustrate this simple case, we do not transform either an opinion query nor the memory itself.

As illustrated in Figure 5.3, the process of opinion formation is a series of two walks over the DTM: The first walk begins with state X_1 whose query is $A_1 = 0, A_2 = 1, A_3 = 0$ and the second walk begins with state X_3 whose query is $A_1 = 0, A_2 = 1, A_3 = 1$. As we can see from the figure, states X_1 and X_3 are two steps away: one can walk from state X_1 to X_3 by the transformation sequence $r(a_1)a(a_2)$ (a_1 denotes $A_3 = 0$ and a_2 denotes $A_3 = 1$) and can walk from state X_3 to state X_1 by the transformation sequence $r(a_2)a(a_1)$. In addition, $r(a_1)$ denotes a deletion rule that removes a_1 from a query and $a(a_2)$ denotes an insertion rule that adds a_2 to a query. We have demonstrated how an opinion request gets translated to a set of opinion queries determining the series of walks over of a DTM, now we

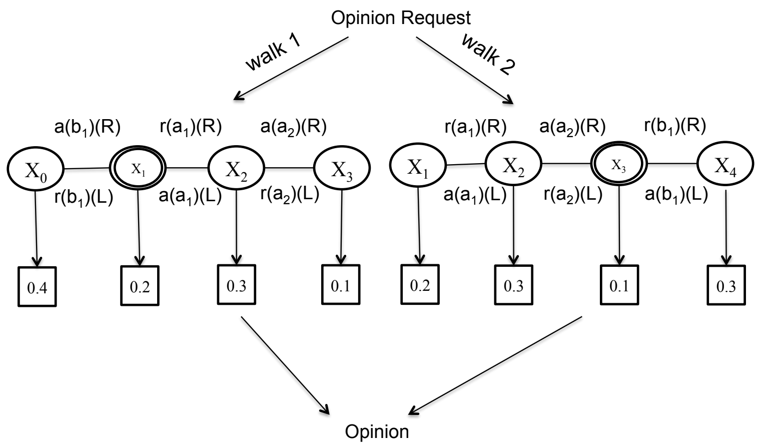


Figure 5.3: Process of Opinion Formation over a DTM

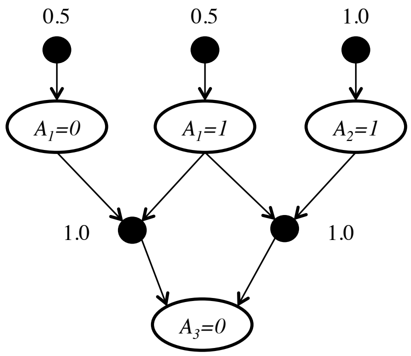
describe how a working memory is formed.

First, we show how to convert **L** into Bayesian Knowledge Bases. The construction of a BKB representing an agent's learning experience is a two-step process. First we convert the vector representation of each learning episode into a BKB representation. Figure 5.4 shows the individual feature-rooted BKBs over feature A_3 for each learning episode in the Tower Room. The second step is to fuse all these BKBs. Figure 5.5 shows the fused BKB which is the base knowledge K.

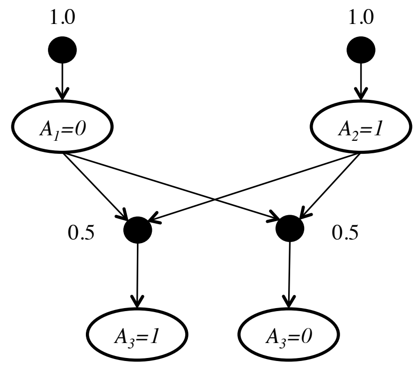
K is the base memory that is formed by encoding all learning episodes which is the domain knowledge of state X_1 and X_3. The probability emitting from X_1 and X_3 are:

$$P(X_1) = 0.21875 \text{ and } P(X_2) = 0.109375$$

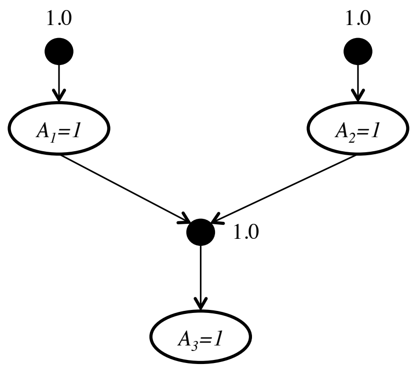
Thus, the opinion distribution for (X_1, X_3) is $(0.21875, 0.109375)$. In the tower room, agents α, δ, and η infer that there is a higher probability that the value for feature A_3 is 0 - this is the opinion they will form and communicate to agent β.



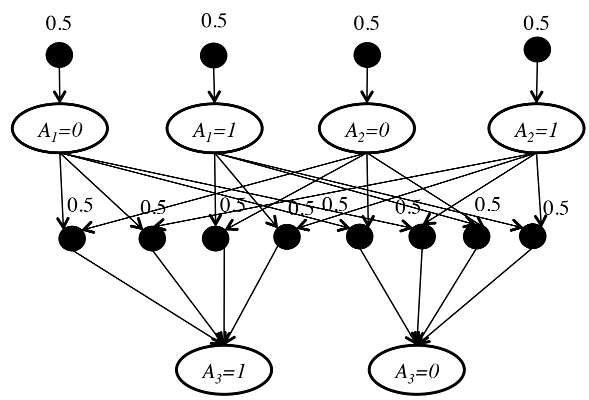
(a) $K^{A_3}_{\mathbf{r_1}}$



(b) $K^{A_3}_{\mathbf{r_2}}$



(c) $K^{A_3}_{\mathbf{r_3}}$



(d) $K^{A_3}_{\mathbf{r_4}}$

Figure 5.4: Feature-Rooted BKBs over Feature A_3 for Different Learning Episodes

5.2.2 Task Transformation

In this section, we demonstrate some examples of task transformation via 1) query transformation, and 2) memory transformation. To make the examples more meaningful, we examine the results of various task transformations with implications for opinion diversity.

We continue with the sample toy problem illustrated in this chapter. In the last section, we demonstrated how a DTM was trained from a collection of learning episodes with capabilities to answer probabilistic questions. To recap, as the objective of a DTM is to model human reasoning behavior in delivering an opinion, we notice the importance of comprehending questions posed by humans expressed through

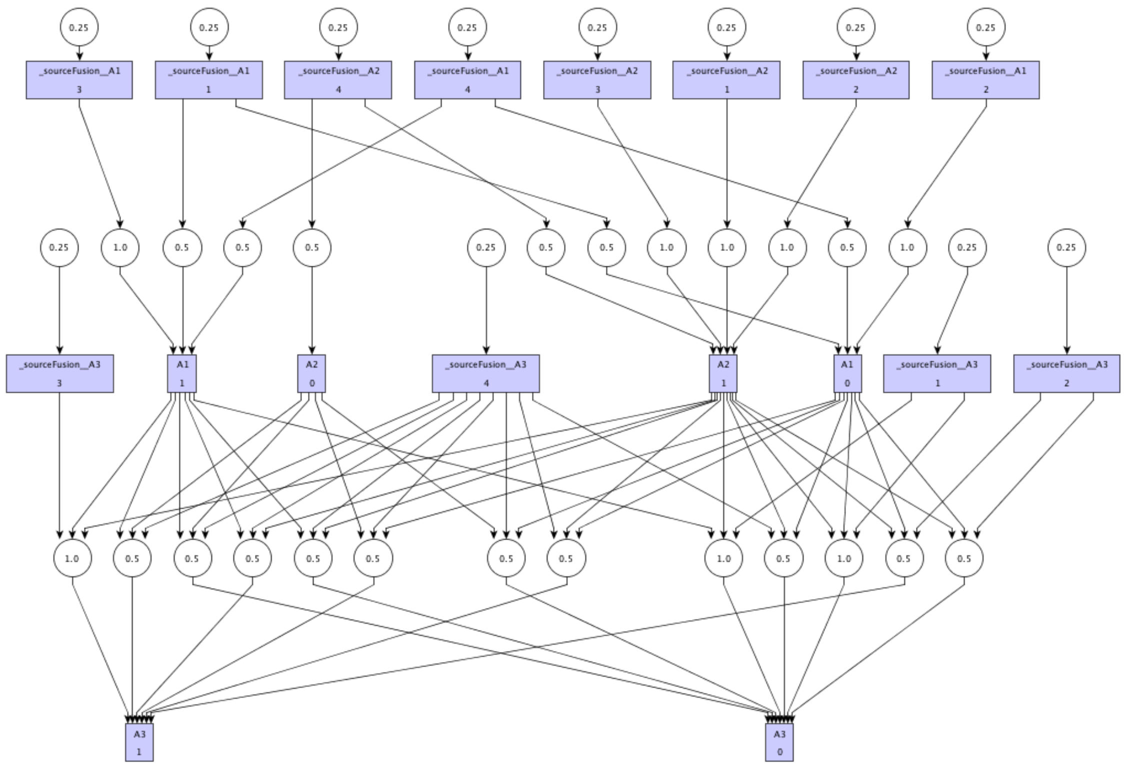


Figure 5.5: BKB Representation K of $\mathbf{L}$

natural language. Therefore, we provided an approach that translates human opinion needs into opinion queries that a DTM understands. In particular, we proposed to describe an opinion request in the form of propositional logic which closely assembles how human express their logic. An opinion request then gets translated into a set of opinion queries signaling how inferences should be done inside a DTM. Since human decision-making behavior is an inexact science, we provide a default decision criteria based on probability results from answering the sequence of opinion queries in a DTM. However, the default decision criteria is open for scholars to investigate its relation to the final opinion.

The novelty of a DTM is its capability of modeling task transformation in opinion formation. In this section, we demonstrate how query and memory transformation may result in *opinion shift*, which can be viewed as a type of *opinion change* from the original opinion that is never formed and communicated externally.

Query Transformation

In this section, we demonstrate query transformations with regards to different "sandboxes". These two imaginary sandboxes have only one object in difference. The Tower Room in the sample toy problem is referred to as the first sandbox s_1 and let us imagine the other sandbox s_1' consists of experts from World Wide Web (we call it a Virtual Room here). As shown in Table 5.3, agents in s_1 can perceive o_1 but cannot see o_9 while agents in s_2 can perceive o_9 but not o_1.

Table 5.6 shows some common query transformations that can occur in fulfilling agent β's question: "What is the value for feature A_3 of object o_5 with observations that $A_1 = 0$ and $A_2 = 1$?". The 1^{st} row (r_1) in Figure 5.6b shows the reasoning result of the opinion model with no query transformation nor knowledge transformation. Thus, an agent in the Tower room may conclude that value for feature A_3 of object o_5 is 0. In Figure 5.6a, rows r_2 to r_4 show the query transformation by applying the rule of evidence deletion. In r_2 and r_3, only one observation of object o_5 is involved in reasoning and no observations is involved for case r_4. However, even though different questions are asked, it is unlikely that an agent shifts its opinion as the probability value for value 0 is still significantly higher than that of value 1.

In cases r_1 to r_4, the query is *feature-driven* in the sense that the query retrieves knowledge related to perceived evidence. In cases r_5 to r_8, the queries become *episode-driven* in the sense that a query recalls one past episode containing specific knowledge rather than general knowledge aggregated from a collection of episodes. By specifying the index of an episode for all features, only the information of that

Table 5.3: Two Different Sandboxes

Room	Learning Episode	A_1	A_2	A_3
Tower Room	o_1	?	1	0
	o_2	0	1	?
	o_3	1	1	1
	o_4	?	?	?
Virtual Room	o_2	0	1	?
	o_3	1	1	1
	o_4	?	?	?
	o_9	?	0	0

	A_1	A_2	A_3	SA_1	SA_2	SA_3
r_1	0	1				
r_2	0					
r_3		1				
r_4						
r_5				1	1	1
r_6				2	2	2
r_7				3	3	3
r_8				4	4	4
r_9	1	0				
r_{10}	1					
r_{11}		0				

(a) 11 Reasonings with Different Queries

	$P(A_3=0)$	$P(A_3=1)$	# worlds for $A_3=0$	# worlds for $A_3=1$
r_1	0.21875	0.109375	36	24
r_2	0.226562	0.117188	39	27
r_3	0.382812	0.273438	60	48
r_4	0.398438	0.289062	66	54
r_5	0.015625	0	2	0
r_6	0.0078125	0.0078125	1	1
r_7	0	0.015625	0	1
r_8	0.0078125	0.0078125	4	4
r_9	0.0078125	0.0078125	3	3
r_{10}	0.171875	0.171875	27	27
r_{11}	0.015625	0.015625	6	6

(b) Reasoning Results

Figure 5.6: Opinion Formation for Agents in Tower Room

episode gets involved in the reasoning. For example, row r_5 inferences over the 1^{st} episode by setting all the source I-nodes (SA_1, SA_2, SA_3) to be 1. As shown in the results, if memory for the 3^{rd} episode is relied upon for inferencing, an opinion shift occurs since $P(A_3 = 1) > P(A_3 = 0)$. The results for the other reasoning over a single episode have equal probability meaning that an opinion is undecided with information from one single episode.

With the capability of query transformation, it is possible to evaluate the impacts of an undesired mistake/inaccuracy. The rows r_9 to r_{11} demonstrate the cases where the perceived observations are incorrect. The results have shown some interesting findings: the opinions become undecided for all cases with incorrect observations. The finding has two implications: 1) the DTM trained from learning episodes gathered in the tower room is *robust* to errors in perception, and 2) this DTM has *low dependency* on observations.

Among 11 common cases of query transformation, 5 of which result in no opinion shift, 5 of which result in undecided opinion, and 1 of which results in a possible opinion shift. In summary, the examples have revealed some interesting phenomena such as: 1) incorrect information may not result in opinion shift (cases r_9 to r_{11}), 2) opinion shift occurs when the information relied upon tells another story (case r_7), 3) more information may not necessarily indicate better results (cases r_2 to r_4), and, 4) an individual can be biased by learning episodes ($P(A_3 = 1)$ is rarely higher than $P(A_3 = 0)$ according to a sample of 11 cases).

Now let us discuss possible opinions formed by the agents in the Tower Room with only query transformations. If we assume these 11 query transformations are chosen with equal chance and the chance of picking any other query transformation is 0, we can calculate the following:

1. The chance that three agents all end up with an opinion shift is 0.17%.
2. The chance that three agents have no opinion shift is 21.33%.
3. The chance that one or more agents ends up with an opinion shift is 14.68%.
4. The chance that one or more agents are undecided is 73.37%.

The results have revealed that: even though the chance of group think (every member delivers the same incorrect opinion) is low (0.17% for this case), the chance of having members with undecided opinions are high. When the size of the group is small, it raises concern of how to aggregate opinions supplied from an individual member.

Now let us look at the individual opinions formed in the Virtual Room. The agents in the virtual room also have access to o_2, o_3, o_4, and additionally a new object o_9 (The BKB representation of o_9 is shown in Figure 5.7a). However, the agents do not have access to object o_1. Conflicting observations can be perceived from objects o_1 and o_9, where the value for feature A_2 is 1 for o_1 while it is 0 for o_9. However, the value for the target feature A_3 is the same for these two objects.

We conducted the same 11 experiments where the query is transformed by setting different evidences. As shown in the results (Figure 5.7b), 3 among them result in an undecided opinion, 3 result in opinion shift (from opinion $A_3 = 0$ to $A_3 = 1$), and 5 result in no opinion shift. Due to different objects that agents in each room have access to, their opinions can end up to be quite different from each other. Compared to the results derived for the Tower Room, the following findings can be drawn from query transformations:

1. *robustness to perception error*: lower compared to agents in the Tower Room.

2. *dependency on observation*: higher compared to agents in the Tower Room.

3. *stability of opinion*: lower compared to agents in the Tower Room.

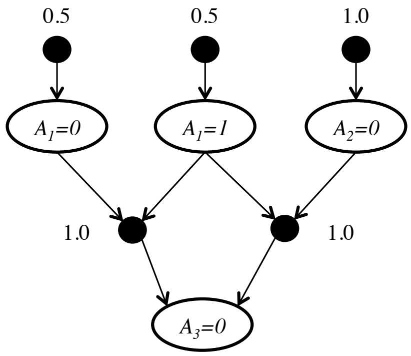
In addition, we also compare group performance for the Virtual Room. If we assume there are also three agents in the virtual room, then we can derive the following:

1. The chance that three agents all end up with an opinion shift is 5.21%.

2. The chance that three agents have no opinion shift is 24.13%.

3. The chance that one or more agents end up with an opinion shift is 50.97%.

4. The chance that one or more agents are undecided is 47.49%.



(a) BKB Representation of Object o_9

	$P(A_3=0)$	$P(A_3=1)$	# worlds for $A_3=0$	# worlds for $A_3=1$
r_1	0.078125	0.078125	18	18
r_2	0.148438	0.273438	30	24
r_3	0.226562	0.117188	39	27
r_4	0.398438	0.289062	66	54
r_5	0.015625	0	2	0
r_6	0.0078125	0.0078125	1	1
r_7	0	0.015625	0	1
r_8	0.0078125	0.0078125	4	4
r_9	0.0703125	0.0234375	12	6
r_{10}	0.109375	0.140625	21	24
r_{11}	0.140625	0.046875	24	12

(b) Reasoning Results

Figure 5.7: Opinion Formation for Agents in Virtual Room

We can see that the chance of having at least one agent delivering an alternative opinion is dramatically higher (14.68% in Tower Room and 50.97% in Virtual Room). We can draw a preliminary conclusion that the members in the Tower Room are wiser than the members in the Virtual Room as the ones in the Tower Room have more similar objects (compare to the target object o_5) to observe.

Memory Transformation

In this section, we demonstrate two simple scenarios of knowledge transformation with the sample toy problem. The first scenario explores the impacts of learning episodes and the second scenario explores the effect of *recency* factors in memory.

In the sample toy problem, we have in total four objects in each room. In the previous section, we were curious about whether including more observations about the target object would increase that chance of delivering a reliable opinion. On the other hand, the same question is concerned with whether fewer observations would result in opinion shift and even group think. In this section, we can explore another question that is often investigated by researchers: Would more training samples increase the chance of delivering a reliable opinion?

In the first scenario, we conduct knowledge transformations where two learning episodes (learning episode regarding o_1 and o_3) are eliminated. It is intuitive from a human perspective that elimination of learning episodes would speed up the process of delivering an opinion. We conduct nine experiments each with a different query transformation (the observation included in each query is shown in Figure 5.8a). As shown in Figure 5.8b, the number of worlds involved in the reasoning with two objects is considerably smaller than the number of worlds involved in the reasoning with four objects. Despite with the fact that only half of the original learning episodes are included, we can observe from the results that only one query transformation results in opinion shift (case r_6). When reasoned with the same query, the opinion formed with four learning episodes and the opinion formed with two are the same. The finding may seem counter-intuitive, but it also has implication that: If the *right* information is retrieved, it can speed up the process of forming an opinion without trade-off with performance.

Another interesting finding is that *less* information considered in the reasoning may *increases* one's uncertainty of his opinion. As shown in Figure 5.8b, for four of nine reasonings, there are reasoning results only for one value of the feature. Therefore, one may feel certain about his opinion as he is only able to see one side of the coin. Furthermore, we would like to point out that the proposed DTM may not yield any results if the information considered in the reasoning does not match up with the observations of the target feature (cases r_9 and r_{11}).

The second scenario demonstrates how knowledge transformation can be used to explore the effects of *memory recency*. In order to compare results, we also

	A_1	A_2	A_3	SA_1	SA_2	SA_3
r_1	0	1				
r_2	0					
r_3		1				
r_4						
r_5				1	1	1
r_6				2	2	2
r_9	1	0				
r_{10}	1					
r_{11}		0				

(a) 9 Reasonings with Different Queries

	$P(A_3=0)$	$P(A_3=1)$	# worlds for $A_3=0$	# worlds for $A_3=1$
r_1	0.125	0	2	0
r_2	0.125	0	2	0
r_3	0.5	0.375	6	4
r_4	0.5	0.375	6	4
r_5	0.125	0	2	0
r_6	0	0.125	0	1
r_9			0	0
r_{10}	0.375	0.375	4	4
r_{11}			0	0

(b) Reasoning Results

Figure 5.8: Memory Transformation - Case 1

	A_1	A_2	A_3	SA_1	SA_2	SA_3
r_1	0	1				
r_2	0					
r_3		1				
r_4						
r_5				1	1	1
r_6				2	2	2
r_9	1	0				
r_{10}	1					
r_{11}		0				

(a) Results with Different Queries

	$P(A_3=0)$	$P(A_3=1)$	# worlds for $A_3=0$	# worlds for $A_3=1$
r_1	0.02	0	2	0
r_2	0.02	0	2	0
r_3	0.2	0.72	6	4
r_4	0.2	0.72	6	4
r_5	0.008	0	2	0
r_6	0	0.512	0	1
r_9				
r_{10}	0.18	0.72	4	4
r_{11}				

(b) Reasoning Results

Figure 5.9: Memory Transformation - Case 2

consider only two objects (o_1 and o_3) for this experiment. In addition to knowledge transformation in eliminating two learning episodes, we also substitute the knowledge regarding the importance measure of an episode. Recall that in the original base memory, equal weights have been assigned to all the source I-nodes meaning that each episode is equally important. Here, we substitute these S-nodes pointing to prior I-nodes with new ones encoding how recent each episode is. Assume that an agent learns about object o_3 after it learns about object o_1, we substitute the S-nodes pointing to I-nodes $SA_1 = 1$, $SA_2 = 1$, and $SA_3 = 1$, with new S-nodes with weights 0.8. Furthermore, we substitute the S-nodes pointing to I-nodes $SA_1 = 3$, $SA_2 = 3$, and $SA_3 = 3$ with new S-nodes with weights 0.2.

We conduct the same 9 experiments on the opinion models with transformed knowledge, the results of which are shown in Figure 5.9a and Figure 5.9b. Among nine experiments, four result in opinion shift, two yields no results due to insufficient information, and three result in no opinion shift. The results from this scenario imply that the recency of an episode can be *misleading* to humans. In particular, memories of objects with similar values for features that are perceived earlier can be faded out quicker and even fail to be recalled and included into working memory. Due to such reasons, opinion shifts may occur by relying upon on the more recent learning episodes. Furthermore, in the cases with opinion shifts, there is a significant distance between $P(A_3 = 1)$ and $P(A_3 = 0)$ meaning that agents do not only result in opinion shifts and they are possibly quite certain with their opinions also.

5.3 Experimental Results

This chapter validates our method of modeling opinion formation by modeling the prevalent heuristics that humans have adopted in reasoning. To begin with, we describe studies about these heuristics and then describe how each heuristic is formulated in our model.

5.3.1 One-Episode Heuristics

The family of one-episode heuristics only rely on one past experience in the process of inferencing; that is, the result is independent of other past experiences. In this work, we cover one type of one-episode heuristics: *Recognition*. In brief, Recognition recalls one past episode referring to the same entity as in the one currently examined.

Recognition

Considering an alternative-question that asks one to compare two cities: Which city, *Hong Kong* or *Kai Feng*, has a larger population? The Recognition heuristics work as follows: If one of the two objects is recognized and the other is not, then infer that the recognized object has the higher value (Gigerenzer and Brighton, 2009). If someone has never heard of the city Kai Feng but has heard of Hong Kong, he may consider Hong Kong to have a larger population. Even though the heuristic sounds overly-simplified, it often works better than random guessing under the circumstances where there are insufficient learning experiences.

The recognition heuristic for a pair-wise comparison can be considered as a series of two walks over an instantiated Double Transition Model. The purpose of the first walk is to evaluate the probability of a recalled episode referring to object o_1 and the second walk is to evaluate the probability of a recalled episode referring to object o_2. The shortest paths in these two walks are as follows:

$$w_1 = a(SA_1 = idx(o_1)) \dots a(SA_n = idx(o_1))$$

$$w_2 = a(SA_1 = idx(o_2)) \dots a(SA_n = idx(o_2))$$

where each r.v. with prefix SA is an I-node denoting the source of the S-node it points to. Here, each walk starts with a state in which the query is empty and domain knowledge is the base memory. $idx(o_1)$ and $idx(o_2)$ are functions to retrieve the index of a learning episode involving that object. If the index of an episode involving the desired object cannot be found, we will have an index valued -1. By fully specifying the sources of all features, the transformed query allows us to retrieve one learning episode from the entire memory.

We may consider such query transformations as a simulation of *memory recall*. All learning episodes are stored in the memory and the individual memory is recalled based on recall cues. (Here, the recall cue is the object involved in a learning episode.) The functions $idx(o_1)$ and $idx(o_2)$ implemented have a perfect matching between an episode and an object. In other words, the function will never fail to recall an episode given an object. Similarly, the function will never recall an episode that mentions the wrong object.

According to the recognition heuristic, the city being recalled is a positive indicator of it being larger. Thus, the heuristic ignores the observation associated with the city. It is possible that both of the two cities in the questions are small and an individual remembers one city being small. However, as he has not heard of the other city, he may draw a conclusion that the other city is even smaller. When we model this heuristic, as the actual observations for the feature *population* are ignored, there is no need to translate an opinion query into multiple queries each specifying one possible observation. In the perspective of probability computations, it can be viewed such that the probability without specifying the state of the target feature is the sum of probabilities of all worlds each with a different state of the target feature.

The approach to model the recognition heuristic can also be generalized to comparisons between multiple objects. Formally, to compare a group of objects $o_1, \ldots, o_k$, we conduct a series of k walks over the DTM.

$$w_1 = a(SA_1 = idx(o_1)) \ldots a(SA_n = idx(o_1))$$

$$\ldots$$

$$w_k = a(SA_1 = idx(o_k)) \ldots a(SA_n = idx(o_k))$$

A simple decision criteria for the Recognition heuristic looks as follows:

$feature(o_1) > feature(o_2)$ if $P(w_1) > 0$ and $P(w_2) = 0$

$feature(o_1) < feature(o_2)$ if $P(w_2) > 0$ and $P(w_1) = 0$

However, the decision criteria can be relaxed to be:

$feature(o_1) > feature(o_2)$ if $P(w_1) > P(w_2)$

$feature(o_1) < feature(o_2)$ if $P(Q_2) > P(w_1)$

A decision criteria is the procedure that finalizes an opinion from an opinion distribution. As we have described earlier, an opinion distribution is a probability distribution for different queries. Each query can be understood as an evaluation on one "aspect" that needs to be examined. Intuitively, an opinion distribution can be of small size as one may only evaluate a small number of aspects but can also be of large size indicating a more careful enterainment of aspects.

With an opinion distribution, one can form a final opinion using different criteria. As illustrated for this heuristic, one may conclude that Hong Kong has a larger population than Kai Feng if he has heard of one but not the other. However, he may relax his criteria by comparing the probabilities of these two cities. In this case, the city with a higher probability of having a population is considered to have a larger population than another one.

5.3.2 One-Feature Heuristics

The family of one-feature heuristics only relies on one feature in the process of inferencing; that is, the result is independent of the values of other features. In this work, we cover three types of heuristics called *Minimalist*, *Take the Best*, and *Take the Last*

(Brighton, 2006; Gigerenzer and Brighton, 2009; Gigerenzer and Gaissmaier, 2011). In brief, minimalist randomly picks a feature to reason upon, take-the-best picks a feature following the order of feature validity (prediction accuracy), and take-the-last picks a feature following the order of episode recency.

Take the Best

The take-the-best heuristic relies on one distinguishing feature, at the same time also the one with highest validity in effectively classifying learning episodes. Take-the-best follows the order of validity of features and examines whether the chosen feature distinguishes the given task also. The steps of take-the-best heuristics described in (Gigerenzer and Goldstein, 1999) are:

1. Step 0. If applicable, use the *recognition heuristic*; that is, if only one object is recognized, predict that it has the higher value on the criterion. If neither is recognized, then guess. If both are recognized, go on to Step 1.

2. Step 1. Ordered Search: Choose the feature with the highest validity that has not yet been tried for this task. Look up the feature values of the two objects.

3. Step 2. Stopping Rule: If one object has a positive cue value ("1") and the other does not (i.e., either "0" or unknown value) then stop the search and go to Step 3. Otherwise go back to Step 1 and search for another cue. If no further cue is found, then guess.

4. Step 3. Decision Rule: Predict that the object with the positive cue value has the higher value on the criterion.

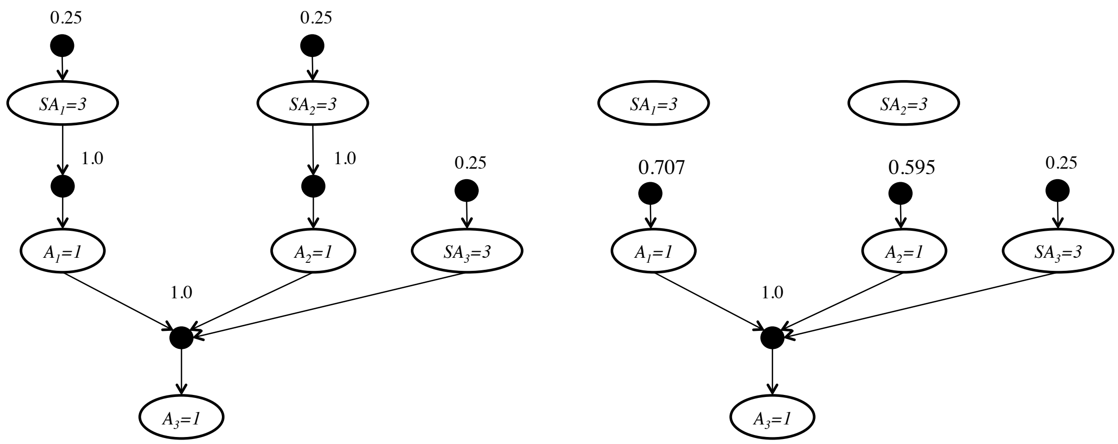
The *validity* of the i^{th} feature is computed as the following:

$$v_i = \frac{\text{number of times feature } i \text{ makes a correct inference}}{\text{number of times feature } i \text{ discriminates between objects}}$$

Given a pair of objects, a feature is said to discriminate between the objects if two objects have different values for this feature. For a feature to discriminate correctly, the object which has the higher criterion value must also have a feature value representing presence of the property represented by the feature. In simple terms, the validity of a feature can be thought of as a measure of how many correct inferences are made using this feature alone.

Within the base memory, the probability of a value of a feature (except the target feature), e.g. $P(A_1 = 0)$, reflects the frequency of seeing feature A_1 to be 0 in all the possible worlds. In the earlier chapters, we have shown that $P(A_1 = 0), \ldots, P(A_1 = |\mathbb{C}|)$ matches the counting of each possible value of feature A_1 in the learning episodes. The probability encodes frequency as an *importance* measure for a feature; that is, the more frequently an observation is perceived, the more dominant this observation is. However, under some situations other types of important measures can be more effective in delivering an opinion. The validity measure of each feature adopted by take-the-best heuristic is an example of such other importance measures. When comparing two or many objects, an observation may have high occurrence, but the feature may not distinguish different objects at all. Validity directly encodes how effective a feature is in distinguishing these objects.

As we have described in Chapter earlier, each auxiliary knowledge represents knowledge induced from the base memory of all learning episodes. Here, an auxiliary BKB encodes the validity measure of a corresponding feature. An example of an auxiliary BKB (part) is shown in Figure 5.10. The validity measure can be computed from all the learning episodes by conducting pair-wise comparisons exhaustively. As the BKB is a probabilistic formalization, the validity value cannot be directly used



(a) K_1 (b) K_1'

Figure 5.10: Auxiliary BKB (Take the Best)

as the prior probability of each feature. However, as long as the order of validity is preserved and the results of probabilistic inferences over the modified memory are the same as those would be produced by take-the-best heuristic, it will produce the same answer. Thus, we propose to encode the prior probabilities in each auxiliary BKB according to the following: For an I-node $A_i = a_i$, the weight of S-node b pointing to it is computed as:

$$a^{\frac{1}{2}^{rank(A_i)}-1}$$

where $1 \leq i \leq n$ and a is a constant bigger than 1, $rank(A_i)$ denotes the rank of its validity value among all features.

For instance, if we have three features with validity ranks $1, 2, 3$, their corresponding weights for the S-nodes are $2^{-\frac{1}{2}} = 0.707, 2^{-\frac{3}{4}} = 0.595, 2^{-\frac{7}{8}} = 0.545$ if we let constant $a = 2$. As shown in Figure 5.10b, the prior probability for $A_1 = 1$ is thus modified to 0.707 and the prior probability for $A_2 = 1$ is modified to 0.595.

Take-the-best heuristic for a pair-wise comparison can be considered as a series of two walks over an instantiated Double Transition Model. The first walk aims to evaluate the probability of the first object on a transformed BKB and the second walk aims to evaluate the probability of the second object on the same transformed BKB. The shortest paths in these two walks look like the following:

$$w_1 = subst(K_1, K_1')a(A_1 = a_1)\ldots a(A_n = a_n)$$

$$w_2 = subst(K_1, K_1')a(A_1 = a_1')\ldots a(A_n = a_n')$$

where in walk w_1, the transformation sequence $a(A_1 = a_1)\ldots a(A_n = a_n)$ are formed according to observations of object o_1. In the 2^{nd} walk w_2, the transformation sequence $a(A_1 = a_1')\ldots a(A_n = a_n')$ are formed according to observations of object o_2.

Minimalist

The minimalist heuristic relies on one distinguishing feature, no matter the degree of its validity, to conduct inferencing. The following are the steps of the minimalist heuristic:

- Step 0. If applicable, use the *recognition heuristic*; that is, if only one object is recognized, predict that it has the higher value on the criterion. If neither is recognized, then guess. If both are recognized, go on to Step 1.
- Step 1. Random Search: Draw a cue randomly (without replacement) and look up cue values of the two objects.
- Step 2. Stopping Rule: If one object has a positive cue value ("1") and the other does not (i.e., either "0" or unknown value) then stop search and go onto

Step 3. Otherwise go back to Step 1 and search for another cue. If no further cue is found, then guess.

- Step 3. Decision Rule: Predict that the object with the positive cue value has the higher value on the criterion.

Compared to the take-the-best heuristic, the minimalist heuristic only differs in Step 1. We model the minimalist heuristic as a variation of take-the-best heuristic. The main idea is to add a *perturbation* in the $rank(A_i)$ function to simulate the effect of vague memory where an individual may not have an exact order of validity values.

Then, based on the new validity values with perturbations applied, an auxiliary BKB, say K_2 and K_2', can be constructed according to the new ranks. The rest of the steps are the same as take-the-best heuristic.

$$w_1 = subst(K_2, K_2')a(A_1 = a_1)\dots a(A_n = a_n)$$

$$w_2 = subst(K_2, K_2')a(A_1 = a_1')\dots a(A_n = a_n')$$

where in walk w_1, the transformation sequence $a(A_1 = a_1)\dots a(A_n = a_n)$ are formed according to observations of object o_1. In the 2^{nd} walk w_2, the transformation sequence $a(A_1 = a_1')\dots a(A_n = a_n')$ are formed according to observations of object o_2.

Take the Last

The take-the-last heuristic relies on one distinguishing feature, following the order of recency, to conduct inferencing. Take-the-last heuristic differs from take-the-best only in Step 1, which becomes:

Step 1. Einstellung Search: If there is a record of which cues stopped search on previous problems, choose the cue that stopped search on the most recent problem and has not yet been tried. Look up the cue values of the two objects. Otherwise try a random cue and build up such a record.

We model take-the-Last also via memory transformation. In the take-the-best heuristic, an individual follows the order of validity values of features. In the take-the-last heuristic, an opinion provider follows the order of how recent a feature worked in comparisons. It is reasonable why an individual will prefer a recent successful feature rather than the feature that distinguishes the best. For instance, whether a city having a metro system is the strongest indicator of whether a city has a large population. However, depending upon where the opinion request comes from, the opinion provider may be frequently asked to compare two cities both with no metro systems. The *validity* value of a feature is independent of its *frequency* value; for example, an observation of a feature may only occur once per a hundred times but every time it occurs, it can successfully compare the objects. Thus, the importance measure in take-the-last heuristic can be treated as a hybrid of the validity and frequency value as measuring how important a feature is.

We model take-the-last heuristic via memory transformation by replacing the original frequency-based importance measure of features with a modified validity-based importance measure. Similar with the way we model take-the-best, we first construct an order of how recent a feature worked for a comparison problem. Then, we use the same formulation to compute the new importance measure for a feature in the memory. With the new importance measure, now we construct auxiliary memory, say K_3 and K_3', for memory transformation. We also construct a series of two walks $\{w_1, w_2\}$, each of which evaluating the probability for one object given its observations.

$$w_1 = subst(K_3, K_3')a(A_1 = a_1) \dots a(A_n = a_n)$$

$$w_2 = subst(K_3, K_3')a(A_1 = a_1') \ldots a(A_n = a_n')$$

where in walk w_1, the transformation sequence $a(A_1 = a_1) \ldots a(A_n = a_n)$ are formed according to observations of object o_1. In the 2^{nd} walk w_2, the transformation sequence $a(A_1 = a_1') \ldots a(A_n = a_n')$ are formed according to observations of object o_2.

Chapter 6

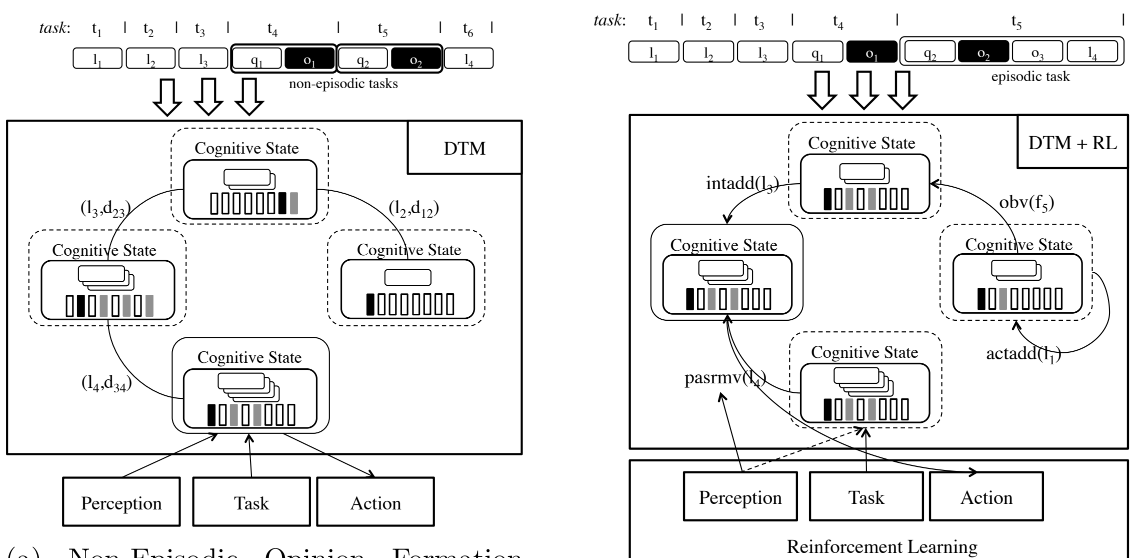
Modeling Opinion Formation with External Influence

This chapter formally describes how to solve episodic opinion formation tasks. Section 1 and Section 2 provide an overview and the motivation for Phase II. Section 3 provides justification for modeling the process of an opinion formation task as a Markov decision process. Section 4 provides formal definitions of a MDP using DTMs and then illustrates five different cases by varying parameters of goal functions in Section 5. Section 6 describes how to solve this task under a stochastic environment.

6.1 Overview

The objective of this section is to provide an overview of Phase II as presented in Chapters 6 and 7. In Phase I, a DTM is capable of conducting a non-episodic opinion formation task which consists of only one opinion formation process (see Figure 6.1a). A non-episodic opinion formation task in Phase I is solved under the assumption of knowing the cognitive state in use. We validated the design of a DTM by modeling four commonly-accepted heuristics in reasoning to show that the design was powerful to derive a space of cognitive states that can cover the ones humans use.

Determining the cognitive state currently in use remains an open question for the research community. Phase II aims to mitigate this problem through modeling episodic tasks. An episodic opinion formation task consists of a sequence of opinion formation and opinion change processes a single issue. We exploit the close relationship between opinion formation and opinion change processes within an episodic task to learn the induction of state(s) in a DTM.



(a) Non-Episodic Opinion Formation Tasks

(b) Episodic Opinion Formation Tasks

Figure 6.1: Non-Episodic (Ch. 4-5) and Episodic (Ch. 6) Opinion Formation Tasks

Chapter 6 solves an episodic opinion formation task (see Figure 6.1b) by defining it as a Markov decision problem: At each time step, determine what is the best action to take to accomplish one's goal. We formulate a goal function that considers two aspects: 1) minimizing the gap between two players' opinions for next time step, and 2) minimizing the change between his own opinions between two steps. It is very important to emphasize here that: an episodic opinion formation task subsumes a non-episodic opinion formation task by defining a finite-horizen MDP with the horizon equal to one. This is a simple yet powerful generalization so that the episodic and non-episodic tasks can now be both modeled as MDPs.

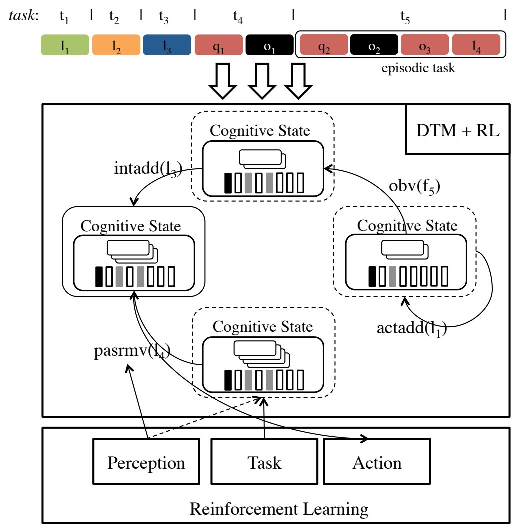


Figure 6.2: Dynamic Opinion Formation Tasks (Ch. 7)

In this chapter, we assume that each individual can fully observe his interlocutor's DTM and has complete knowledge of his interlocutor's reactions. Theoretically, the Markov decision problems we define in this chapter are guaranteed to have optimal solutions. However, the necessary assumption can be too strong under situations where we do not have perfect knowledge of other individual's DTMs in various real-world situations. Thus, we further developed in Chapter 7, a Q-learning method which uses a model-free reinforcement learning algorithm. If training experience is available, a Q-learning method can improve an agent's strategy by repetitively participating in episodic opinion formation tasks.

With approximation methods such as Q-learning methods (see Algorithm 3.5) and exact solutions such as value-iteraction (see Algorithm 3.1) and policy-iteraction methods (see Algorithm 3.2), the framework can handle dynamic opinion formation tasks that consist of a sequence of opinion formation and change processes with external influences from multiple sources. As shown in Figure 6.2, a non-episodic opinion formation task t_4 and an episodic opinion formation task t_5 form a dynamic opinion formation task assuming both these two tasks occur within a short period of time. The framework with DTMs plus reinforcement learning methods as its fundamental building blocks now becomes a complete framework of *computational opinions*.

To proceed, we shall explicitly state the differences between Phase I and Phase II. As the main focus of Phase I is to design a cognitive model compatible with well-recognized theories in bounded rationality, social theories, and in reasoning, we focused on the representational flexibility of the cognitive model. To reduce the complexity of the work in Phase II, we have made the following simplifications:

- Within one opinion formation task, individuals never switch queries.
- Individuals always correctly interprete the query and never modify it during inferencing (e.g. discard an evidence).
- An individual can send up to one message to another individual each time.

6.2 Approach

6.2.1 Motivation

In Chapter 4, we proposed a double transition model as a cognitive model. In a DTM, each node is a cognitive state of reasoning that emits a probability. Each edge within a DTM denotes differences in either knowledge or query between two connected nodes. The choice of a working memory and the choice of a query in forming an opinion is modeled as undergoing a sequence of transitions (e.g. select a heuristic in reasoning to simplify the opinion formation task) starting from the long-term memory and a full specification of the query. We have demonstrated in

Chapter 5 that four known human reasoning heuristics can be formed as a sequence of such transitions. Aside from our approach, such a unifying solution has not yet been found in existing cognitive frameworks.

The modeling challenge for the DTM itself is the difficulty to figure out the actual sequence of transitions that take places to form the initial opinions. In general,

- Which reasoning heuristic is in use?
- Why is a reasoning heuristic chosen over another heuristic?
- What are other possible ways individuals choose to simplify problems?
- How, where, and what with regards to internal and external influents?

The objective of Phase I is to design and construct a DTM that can cover a sufficient and meaningful space of cognitive states, from each of which an opinion can be derived. However, how this state space is searched is not available from the DTM itself. Formally, the challenges we face in Phase I can be addressed by answering this question:

> *How do we know which cognitive state is in use for deriving an opinion? Basically, what is the choice of a working memory k and the choice of a query q?*

This question can be further translated into a question on opinion formation and another question on the mechanisms of opinion formation:

- Assuming we know an individual's opinion o_1, what is the choice of working memory k and the choice of query q?
- Assuming we do not know an individual's opinion o_1, what is the choice of k and that of query q?

Clearly, the second question is harder than the first one. Both questions are essentially a search problem over the state space of a DTM but the second one has a much larger space when compared to the first question.

We hope to better answer these two questions in Phase II. The external influence that results in opinion change has been the focus in the research community especially with evidence having shown that external influence may persist longer than internal influence (e.g. gaining new information through reading) (Hoekstra, 1995; Watts and Dodds, 2007; Watts and Holt, 1979; Zaller, 1992). As the social theories have implied, the search function changes from one formation task to another formation task. If the search functions are the same, then the persistence of opinion change will be present in all situations. Therefore, answering the second problem will significantly help advance the state-of-art in this area.

Unfortunately, this is a really challenging problem itself mainly due to the massive size of the space to search and complexity inherent in human cognitive processes. However, we can address this problem *as a learning problem rather than as a search problem*. The key difference is that a search problem needs a particular target while a learning problem induces one by itself. We realized this after recognizing the learning nature of opinion change and decision-making nature of opinion formation: essentially what has been learned through internalizing an external influence guides how decisions are made by externalizing internal cognitive processes.

Intuition

To tackle this challenging problem, we scope down the problem by focusing on modeling the interplay between opinion change and formation within one formation task. Within each opinion formation task, we assume that we know about the initial opinion (the choice of k and q). If we can model an opinion formation task by truly internalizing external influence which in turn changes how it externalizes the internal cognitive processes, it may provide insights on how we can address this grand problem. The challenges for building a computational framework lies in modeling both the learning and decision aspects for the entire opinion formation task. Our insight to address this modeling challenge is that we recognize that an individual performing an opinion formation task is essentially a sequential decision problem with a goal in mind.

Internal and External Influents

To understand what decisions are made in an opinion formation task, let us first introduce the concept of influents. We specify two categories of influents by their sources: *internal influents* (e.g. re-evaluate possessed knowledge) and *external influents*. Examples of external influents can be messages gained through reading, through debates and discussions and questions posed by another individual, or perception of others' behavior (e.g. abusive, psychotheraputive, etc.). We concentrate on influents in the form of messages. What can be internalized is unclear as it all happens within a human brain that is inaccessible to others. A variety of social theories have identified that the internalization may include the knowledge basis from which an opinion is formed, the value system, sentiments towards entities, and so forth. Here, we concentrate on the underlying knowledge basis and the reasoning process from which an opinion can be derived.

Now, we formally define an internal influent to be any information originated inside a human brain and an external influent to be any information from the environment. Note that an individual can also generate external influents to others (e.g. sending messages to someone else). An influent is a directed flow of a message and its direction refers to the source of the flow - an influent being a message sent from A to B is thus different from an influent sent from B to A even if the content is the same. The concept of influence places emphasis on the changes that are triggered by an influent.

Modeling Episodic Opinion Formation Tasks

An individual can go through an interactive process of delivering his opinions (generating external influents to others) and processing feedbacks (external influents) of the opinions that may yield changes in internal cognitive processes (internal influents). We define this interactive process to be an episodic opinion formation task.

We formally define two tasks of which an individual is asked to provide opinions:

- Non-episodic opinion formation task: An individual is asked to provide his opinion (represented by a probability) on an issue (represented by a query) but no feedback is given.

- Episodic opinion formation task: An individual engages in a sequence of opinion exchanges with another individual. In each round, an individual may receive an opinion from another individual, reports his own opinion, receives a message explaining the opinion, or sends a message explaining his opinion.

These two tasks are both representative of common opinion exchange situations. Asking a librarian's opinion can be considered a non-episodic opinion formation task where one individual tends to be the domain expert and the other one asks for his expert opinions. Note that non-episodic opinion formation tasks can be either goal-driven or goal-free. On the other hand, episodic opinion formation tasks are also common. For example, if two people are estimating the weight of a cow in a photo, they can go through several iterations to describe their estimation as well as their reasons. We describe episodic opinion formation tasks as goal-oriented tasks -

where a feedback is originated by either a goal of minimizing the gap between two individuals' opinions or to confirm the presence of a gap.

To restate, Phase II focuses on modeling episodic opinion formation tasks as the opinions and the message from another individual are external influences. The internalization of external influence resulting in an episodic task will shed light on how an individual conducts an opinion formation task in general. Now the questions is: How to model an episodic opinion formation task that captures the learning nature of opinion change and decision nature of opinion formation given that we focus on influents in the forms of messages?

6.3 Markov Decision Problem

By explicitly modeling internal and external influents, the edges in a DTM can be exploited to model how an opinion formation process gets induced and affected. In Phase I, a DTM does not differentiate *causes* of either query or memory transitions. We were inspired to convert these undirected edges into directed ones to differentiate the source of stimulus that may trigger a transition from one state to another.

As we have described earlier, an individual performing an opinion formation task is essentially solving a sequential decision problem with a goal. We describe the decision problem in an episodic opinion formation task between two persons as follows:

> *Decision Problem*: Two agents are exchanging influents with each other guided by their respective goals. At each time step, an agent e_1 needs to decide an action to take.

The goal function for e_1 can be defined as

$$\lim_{t \to +\infty} \{\gamma_1 |o_1^{t+1} - o_1^t| + \zeta_1 |o_1^{t+1} - o_2^{t+1}|\} = 0$$

where γ_1,$\zeta_1 \in [0, 1]$ are control parameters. The first term represents the degree of opinion change between time t and $t+1$ for e_1 while the second term represents the gap in two agents' opinions at time $t+1$.

Similarly, the other agent e_2 also has a goal function defined as

$$\lim_{t \to +\infty} \{\gamma_2 |o_2^{t+1} - o_2^t| + \zeta_2 |o_1^{t+1} - o_2^{t+1}|\} = 0$$

where γ_2,$\zeta_2 \in [0, 1]$ are control parameters. The first term represents the degree of opinion change between time t and $t+1$ for e_2 while the second term represents the gap in two agents' opinions at time $t+1$.

The goal function is defined by the needs to cover two possible ways to reduce the gap between two agents: one by moving e_1's opinion towards e_2's and other way by moving e_2's opinion towards e_1's.

What are the meanings of the two parameters γ_1 and γ_2? Intuitively, γ_1 and γ_2 are on a *malleability-idealism scale* from 0 to 1 representing an agent's willingness to change its own opinion; while parameters ζ_1 and ζ_2 are on a *passivity-activism scale* from 0 to 1 representing an agent's eagerness for reaching a consensus. The higher the malleability-idealism score is, the more idealistic an agent is (i.e., more unwilling to change its opinion). The higher the passivity-activism score is, the more active an agent is (i.e., more eager to reach a consensus). We can thus characterize an agent by four different canonical styles in pursuing its goal by setting γ and ζ as follows:

- Idealistic-Active (abbrv. IA) Style: $\gamma_i = 1$ and $\zeta_i = 1$ that $i \in \{1, 2\}$.

  Agent e_i does not change its opinion but changes the other agent's opinion to match his own.

- Malleable-Passive (abbrv. MP) Style: $\gamma_i = 0$ and $\zeta_i = 0$ that $i \in \{1, 2\}$.

  Agent e_i has no particular goal.

- Idealistic-Passive (abbrv. IP) Style: $\gamma_i = 1$ and $\zeta_i = 0$ that $i \in \{1, 2\}$.

  Agent e_i does not change its opinion and does not care to change the other agent's opinion.

- Malleable-Active (abbrv. MA) Style: $\gamma_i = 0$ and $\zeta_i = 1$ that $i \in \{1, 2\}$.

  Agent e_i changes opinion to match the world.

Among these four types of agents, an IP-style agent has little interests in persuading others and in modifying its own opinion. We can consider IP-style e_1 to be a reference of opinion; that is, other agents in the environment can observe its opinion which is stationary over time. An IA-style agent is interested in changing others' opinion but has no interests in changing its own. On the other hand, a MA-style agent exhibits an opposite preference compared to an IA-style agent; it is keen to be adaptive but has no interests in changing others' opinion. A MP-style agent has no particular preferences on the changes of two agents' opinions.

It is useful to explain why the goal function is defined at the level of individuals rather than at the group level (in this case, a group has two agents). Primarily, the group level goal (e.g. finds best actions to reduce the gap) is only instructive to individuals but cannot dictate how exactly each agent accomplishes this goal. For instance, the goal of a medical appointment held between a doctor and a patient is to reach an informed decision on how to proceed with the patient's case. There is a general goal in place (even if not explicitly stated) that the entire medical team shoots for. However, the exact goal in each individual's mind is neither demanded nor planned in advance leaving it unclear how an individual establishes his/her own goal. Briefly speaking, both the patient and the doctor can either choose to convince the other or choose to adjust himself. Therefore, we choose to model the problem on the level of the individuals as it is clearer with less ambiguity.

At first, we focus on the goal functions at defined above. However, this does not need to be the only choice. Later in this section, we will show that any goal function that is defined in terms of DTMs can be used in defining a MDP. We stick to the above goal function for now in order to demonstrate why a MDP is a natural choice for modeling this decision problem.

The goal function is used to assess how desirable each transition between states is for an agent. An agent with high malleability-idealism score prefers transitions with small changes in its own opinion while an agent with high passivity-activism score prefers a transition into a state with smaller opinion gaps. As both agents' opinions are used in evaluating the goal, a *state* at a time step needs to include both agents' opinions. In order to transition from one state to a desired state according to the goal, every agent needs to decide on the actions to take. As described in the previous section, we simplify the problem that each external influent contains one message (corresponds to a learning episode in our framework). Therefore in Phase II we construct a DTM with states connected only if the difference between their domain knowledge is one learning episode. Therefore, each action is an influent containing one learning episode. A state of the environment can be as simple as a pair (two agents' opinions) or as complex as a 4-tuple (two agents' opinions plus a message each agent wants to send).

Next, we consider the *scope of actions* e_1 can take. Which knowledge can it communicate with others? Where do we obtain this knowledge? How realistic is the scope of actions? *The relationship between the scope of actions and the cognitive basis from which an opinion gets formed is the main challenge in modeling opinion formation with external influence.* To tackle this challenge, we imagine a maze problem where an agent identifies the actions it can take by observing the room it is in: checking the doors that are unlocked, examining the ceilings and floors for hidden pathways, and even considering digging a hole through a thin wall.

We thus define a *state-action function* that maps each state to a set of actions that e_1 can take. We assume that the scope of actions is independent of the other agent's opinion; that is, we will have a state-action function for each agent, formally defined as $A_1(s_t)$ and $A_2(s_t)$. A DTM now defines the state-action function, as follows

$$A_1(s_t) = A_1(D_1^t)$$

where s_t is the state of the environment at time t and D_1^t is the current cognitive state in e_1's dtm.

$$A_2(s_t) = A_2(D_2^t)$$

where s_t is the state of the environment at time t and D_2^t is the current cognitive state in e_2's DTM. Intuitively, if one learning episode say l is not in e_1's current working memory K_1^t, how can it say anything about l? Similarly, if l is already in e_1's current working memory K_1^t, accepting l from another agent may not result in any opinion change[1]. Thus we can see how a DTM constructed from a collection of perceptions over time now provides a good basis of how the state-action function

[1] This is an assumption made here. It is possible that the importance of a message gets reinforced over time which increases the persistence of opinion change in the long run.

may look like.

Up to this point, we have illustrated the main components that form a decision-making process for agent e_1 to solve the problem:

1) e_1's goal function to evaluate the desirability of a state.

2) States of the environment (can be in the form of a 2-tuple, 3-tuple, and 4-tuple).

3) e_1's state-action function that determines the set of actions e_1 can take.

4) e_2's goal function to evaluate the desirability of a state.

5) e_2's state-action function that determines the set of actions e_2 can take.

The problem now can be solved in the following way: Starting from an initial state (assuming both agents' opinions are observable to agent e_1), agent e_1 first identifies all the actions it can take from this state. By an accurate estimation of what action e_2 would perform, e_1 can identify all the states that it can transit to. Finally, agent e_1 selects the action that leads to the most desirable state as evaluated by its goal function. This solution requires some knowledge of the other agent (component 4) and 5) in the list above).

What we have derived up to this point is precisely a Markov decision problem. In the next section we provide formal definitions of Markov decision processes derived from DTMs and goal functions.

6.4 Definitions

We formally present our methodology in this section - modeling a Markov decision process (abbrv. MDP) for e_1 based on two agents' DTMs and goal functions. It is important to clarify that we are solving agent e_1's decision problem, not to solve agent e_2's decision problem. As we described in the previous section, we are neither solving a cooperative nor collaborative decision problem on the level of a group. Instead, each agent has its own goal function which is independent of how the other agent obtains its goal function. An interpretation of agent e_2 is that it is part of the environment for agent e_1.

To define a MDP for e_1, e_1's DTM defines the scope of its own knowledge-based behavior while e_2's DTM is e_1's knowledge-based behavior model of e_2. By assuming that e_1 has complete knowledge of e_2 (goal function and the DTM), we can directly solve the problem we stated in this chapter by deriving the optimal policy from a MDP. We shall note that with the assumption of complete knowledge (both full observability and complete knowledge of e_2's dynamics), e_1 and e_2 will have identical DTMs.

Both these two assumptions that e_1 has complete knowledge of e_2's DTM and is aware of e_2's goal function may not be very realistic. Under certain situations, e_2's DTM can be extremely hard to obtain due to insufficient understanding of e_2. For example, due to limited communication, a doctor is unlikely to know how a patient would react to certain opinions and what medical knowledge the patient has. On the other hand, the goal function of a patient is likely to be explicitly stated before or elicited during their discussions. In adversarial situations, a complete knowledge of e_2's DTM may be obtainable through intensive studies and repetitive encounters but e_2 may try to hide or disguise his/her goal function. In the next chapter, we describe how we address these two assumptions via model-free reinforcement learning methods.

Here, we formally define an augmented DTM as the first step towards achieving our Phase II goals. The basic idea is to convert an undirected graph into a transition model, and the probability along the edge denotes the validity of an action for transition between two cognitive states.

Definition 1 (Augmented Query Transition Graph)**.** An augmented query transition graph Q is a triple (Q, A^Q, C^Q) where $Q = (V^Q, E^Q)$ is a query transition graph, A^Q is finite set (of actions), and C^Q is a function from $V^Q \times A^Q \times V^Q$ into $[0, 1]$ such that $\sum_{v_2 \in V_Q} C^Q(v_1, a, v_2) = 1$ for all $v_1 \in V_Q$ and $a \in A^Q$.

We shall use the same Q to denote both the augmented query transition graph and the query transition graph involved, if no ambiguity is likely to arise.

In the above definition, $C^Q(v_1, a, v_2)$ represents the transition probability that given query v_1 and action a, the current query will be transformed to query v_2.

Definition 2 (Augmented Memory Transition Graph)**.** An augmented memory transition graph K is a triple (K, A^K, C^K) where $K = (V^K, E^K)$ is a memory transition graph, A^K is finite set (of actions), and C^K is a function from $V^K \times A^K \times V^K$ into $[0, 1]$ such that $\sum_{v_2 \in V_K} C^K(v_1, a, v_2) = 1$ for all $v_1 \in V_K$ and $a \in A^K$.

We shall use the same K to denote both the memory transition graph and the augmented memory transition graph involved, if no ambiguity is likely to arise.

In the above definition, $C^K(v_1, a, v_2)$ represents the transition probability that given domain knowledge in a memory v_1 and action a, the current domain knowledge will be transformed to domain knowledge in v_2.

Definition 3 (Augmented Double Transition Model)**.** An augmented DTM $D = (Q \times K)$ where Q is an augmented query transition graph and K is an augmented memory transition graph. In addition, if Q consists of only one state, then D is a simple DTM. In this case, we shall use K to denote D.

When using MDPs for generating and modifying opinions, it is easy to see how one could obtain an MDP M from a simple augmented DTM K. In this case, each vertex in K corresponds to a state in M, the action set of M is equal to the action set in K, and the transition probability in M is equal to the corresponding transition probability in K. The only item missing for M is the reward function which can directly derived from a given goal function.

The goal function captures how desirable a state is for an agent based on which the utilities (value) of a state can be computed to instruct an agent's action (policy). In this work we define a goal function based on the differences of opinions between two agents, but other variations of goal functions can also be designed to instantiate a MDP problem. For example, a goal function from an adversarial perspective can be that an agent deliberately looks like opinion divergence with another agent - in this case the a state with smaller gap between two agents will have a lower reward value.

In the rest of this chapter, we restrict our attention to the case where only two agents are involved and only certain actions are allowed. Since the goal function for e_1 and e_2 plays an important role in their opinion dynamics, we describe the formulations of a Markov decision process in five situations: 1) MA-MA interaction, 2) IA-MA interaction, 3) MA-IA interaction, 4) IA-IA interaction, and 5) Mixed-goal vs. Mixed-goal interaction. For conciseness, we only provide full details on the MA-MA case in this chapter, details of the other four cases can be found in Appendix C. These situations will help illustrate and demonstrate the representational power of DTMs direct transformations to MDPs.

We specify four attributes for each $a \in A^K$ according to our discussion on external influents in the section "internal and external influents".

- direction: from agent e_i to e_j where $i, j \in \{1, 2\}$
- source: e_i that $i \in \{1, 2\}$
- type: add, remove, and do nothing
- content: a learning episode l_j for $j = 1, 2, \ldots, m$

Since all the attributes are symmetric, i.e., applicable to both e_1 and e_2, in what follows, we shall not distinguish between e_1 and e_2. We specify seven types of actions from these four attributes as the follows:

- intremove(l_j) denotes an internal action to exclude learning episode l_j from working memory.
- intadd(l_j) denotes an internal action to include learning episode l_j into working memory.
- pasadd(l_j,e_i) denotes an action to include learning episode l_j into working memory suggested by the other agent e_i.
- pasremove(l_j,e_i) denotes an action to exclude learning episode l_j from this working memory suggested by the other agent e_i.
- actadd(l_j) denotes an action to suggest the other agent e_i to include learning episode l_j into working memory.
- actremove(l_j,e_i) denotes an action to suggest the other agent e_i to exclude learning episode l_j from working memory.
- donothing.

We consider these seven types of actions in solving deterministic and stochastic tasks in the next section, as well as for solving dynamic tasks in Chapter 7.

6.5 Deterministic Task

Even though we have provided mathematical definitions of MDP based on two agents' DTMs and their goal functions, in this section we demonstrate step-by-step what each ingredient of a MDP looks like and how they are assembled. Furthermore, the MDPs defined for the MA-MA interaction, IA-MA interaction, MA-IA interaction, IA-IA interaction, and Mixed-Goal versus Mixed-Goal interaction are the MDPs used in the experiments we conduct in Chapter 7.

For the purposes of experimentation, we focus on interactions between active agents (MA-style agent or IA-style agent) in Chapter 7. However, other types of agents such as MP-style agent and IP-style agents are also worthy of attention. A MP-style agent can be used to model an individual who is less cognitively involved into an opinion formation task as it does not have any particular desire to reach a consensus with the other agent. In contrast with an IP-style agent, a MP-style agent does not mind changing its own opinion which may result in situations that it is likely to accept messages even if they are conflicting. The MP-style agent fits well for modeling individuals with low cognitive-awareness (Receive-Accept-Sampling model). An IP-style agent does not have the desire to reach a consensus, however it also does not like changing its own opinions. An IP-style agent fits well at modeling stubborn individuals (Yildiz et al., 2011) and closed-mind individuals (Vacchiano et al., 1969).

6.5.1 Malleable-Active vs. Malleable-Active Agent

A malleable-active agent seeks to adapt its opinion to the other's and is not interested in instructing how the other agent should behave. At each time step, the primary MA-style agent makes a decision on the proper knowledge transformation to pursue in order to converge to the other agent's opinion. For instance, it can consider discarding some learning episodes that seem irrelevant, or including information that was previously neglected, or do nothing after re-examining all the information it has collected so far.

When one-to-one communication is limited (e.g. online forums), the decision-making problem can be defined as a MA-MA Markov decision problem. One may analyze the source of information that results in differences in their opinions, and then adapt to the other by reconsidering the information to draw conclusions from.

A state of the environment $s \in S$ is a pair (K_1^t, K_2^t) where K_1^t is the knowledge base relied upon by agent e_1 at time t, and K_2^t is the knowledge base relied upon by agent e_2 at time t. Set S is thus a cross product of set V_1^K and set V_2^K capturing all possible combinations of knowledge agents e_1 and e_2 may use to derive their opinions at a given time. We do not provide full details here but it is straightforward to derive the number of states in which consensus is reached (e.g. $o_1 = o_2$ such that $P(K_1|q_1) = o_1$ and $P(K_2|q_2) = o_2$) or the number of states where opinion divergence is present (e.g. $|o_1 - o_2| < \delta$ where δ is a threshold).

We now consider the seven forms of actions: intadd, intremove, pasadd, pasremove, actadd, actremove, and donothing. As a malleable-active agent does not communicate, we only need to consider intadd, intremove, donothing actions for the MA-MA case. The exact learning episode for an action depends on the learning episodes included in the knowledge base currently in use. Each action functions is independent of the other agent's state

$$A_1(s^t) = A_1(K_1^t)$$

and

$$A_2(s^t) = A_2(K_2^t)$$

where K_1^t is the domain knowledge agent e_1 relies upon at time t, K_2^t is the domain knowledge agent e_2 relies upon at time t.

Then, the *action function* for e_1 and e_2 is defined as

$$a \in A_i(s^t) \text{ such that } i \in [1,2] \text{ if: } \begin{cases} \mathit{type}(a) \text{ is intadd, and } \mathit{value}(a) \notin K_i^t \\ \mathit{type}(a) \text{ is intremove, and } \mathit{value}(a) \in K_i^t \\ \mathit{type}(a) \text{ is donothing, and } \mathit{value}(a) = \phi \end{cases}$$

The derivation of **transition probabilities** are broken down into two parts: the first part considers the validity of a transition for agent e_1 while the second part considers the validity of a transition for agent e_2

$$P(s_t, a_1, s_{t+1}) = \begin{cases} 1.0 \text{ if } P(K_1^t, a_1, K_1^{t+1}) = 1 \text{ and } P(K_2^t, a_2^*, K_2^{t+1}) = 1 \\ 0 \text{ otherwise} \end{cases}$$

where $P(K_1^t, a_1, K_1^{t+1})$ is the probability of resulting in a knowledge base K_1^{t+1} by applying change a_1 to the current knowledge base K_1^t. $P^(K_2^t, a_2^*, K_2^{t+1})$ is the probability of resulting in a knowledge base K_2^{t+1} by applying change a_2^* to the current knowledge base K_2^t. For instance, if agent e_1 does nothing ($\mathit{type}(a_1) = \mathit{donothing}$), then we should have $K_1^t = K_1^{t+1}$. We assume the environment to be non-stochastic (e.g. information-processing produced is always correct); that is, a valid knowledge transformation action has a probability of 1.0. We formally specify the transition probability as the following:

$$P(K_1^t, a_1, K_1^{t+1}) = \begin{cases} 1.0 \text{ if one of the conditions below holds} \\ 0 \text{ otherwise} \end{cases}$$

- $type(a_1) = donothing, K_1^{t+1} = K_1^t$
- $type(a_1) = intadd, K_1^{t+1} = K_1^t \cup value(a_1)$
- $type(a_1) = intremove, K_1^t = K_1^{t+1} \cup value(a_1)$

and

$$P(K_2^t, a_2^*, K_2^{t+1}) = \begin{cases} 1.0 \text{ if one of the conditions below holds} \\ 0 \text{ otherwise} \end{cases}$$

- $a_2^* = \max_{a \in A_2(s^t)}(R_a(s, s')), type(a_2^*) = donothing, K_2^{t+1} = K_2^t$
- $a_2^* = \max_{a \in A_2(s^t)}(R_a(s, s')), type(a_2^*) = intadd, K_2^{t+1} = K_2^t \cup value(a_2^*)$
- $a_2^* = \max_{a \in A_2(s^t)}(R_a(s, s')), type(a_2^*) = intremove, K_2^t = K_2^{t+1} \cup value(a_2^*)$

where a_2^* is the greedy action e_2 decides according to its own goal function.

The *reward function* for e_1 and e_2 is derived from the goal function

$$R_a(s, s') = -|o_1^{t+1} - o_2^{t+1}|$$

where $P(K_1^{t+1}|q_1^{t+1}) = o_1^{t+1}$ and $P(K_2^{t+1}|q_2^{t+1}) = o_2^{t+1}$. Due to our simplifications in Phase II, we focus on one query and do not consider diversifications on the query. Thus, we have $q_1^t = q_2^t$ for all values of t.

Now, we have provided a complete definition of a MDP (states, action function, reward function, and probability transition function) for a MA-MA case. Detailed specification of the key ingredients of a MDP for IA-MA interaction, IA-IA interaction, MA-IA interaction, and Mixed-Goal versus Mixed-Goal interaction can be found Appendix C.

6.6 Stochastic Task

So far, we have considered a purely deterministic case for an episodic opinion formation task. Here, we relax the requirement for determinism and briefly consider a stochastic version we call stochastic opinion formation.

Determinism is achieved in our previous formulations due to the following assumption: Agent e_2's action at time t is uniquely determined based on agent e_1's action at time t and agent e_2's greedy policy (e.g., minimizes $|o_2^{t+1} - o_2^t| + |o_1^{t+1} - o_2^{t+1}|$ for IA). Recall that agent e_1's possible actions are constrained by its DTM D_1 coupled with agent e_2's single greedy choice which results in a unique state transition to s_{t+1}. Interactions only occurred when learning episodes are sent between agents or requests to eliminate learning episodes are made from one agent to the other. As we saw earlier, MA-MA was the case of no interactions except for observing the emitted probability values, o_1^t and o_2^t. Now, instead of a greedy deterministic policy for agent e_2, assume that agent e_2's choice of action with respect to agent e_1's action is probabilistic. Of course, certain action pairs between the agents remain tightly coupled such as one agent can only accept a learning episode if the other agent sends it. Note, when one agent requests the other to remove a learning episode, the other can choose to remove it with pasremove or decide independently to remove with intremove. Still, the effects are the same given the current model definitions in which the states and greedy function are unaffected. Intuitively, this means that for either case, the space of available actions A_1 and A_2 at a given state s_t can be limited with certain actions having 0 probability (as in the former case).

As such, this stochastic version modifies the transition probability function $P(s_t, a_1, s_{t+1})$. It still remains the case that the particular choice of actions for each agent will dictate the outcome state $s_{t+1} = (K_1^{t+1}, K_2^{t+1})$. For example, if $a_1 = \text{intremove}(l)$ and $a_2 = \text{donothing}$, then $K_1^{t+1} = K_1^t - \{l\}$ and $K_2^{t+1} = K_2^t$. However, the action choice for agent e_2 is now probabilistic.

We use $s_t \xrightarrow[<a_1,a_2>]{} s_{t+1}$ to denote the transformation from state s_t applying actions $< a_1, a_2 >$ to state s_{t+1}.

In essence, when in state s_t and agent e_1 has selected action a_1, agent e_2 now probabilistically selects its action a_2. Based on this selection, each a_2 will potentially result in a different outcome state s_{t+1}. As such, the transition probability function $P(s_t, a_1, s_{t+1})$ will be equal to the probability of selection an a_2 that ends up in state s_{t+1}. There is a wide variety of possible stochastic agents that can be formulated. Here we illustrate one using a uniform distribution to select actions.

Let $A_2(s_t|a_1) \subseteq A_2(s_t)$ be the set of possible actions for agent e_2 as constrained by agent e_1 having selected action a_1. This can be readily defined as follows: For all learning episodes l_j for $j = 1, 2, \ldots, m$,

- For $a_1 \in \{\text{intremove}(l_j), \text{intadd}(l_j), \text{donothing}\}$, $A_2(s_t|a_1) = A_2(s_t) - \bigcup_{i=1}^{m} \{\text{pasadd}(l_i, e_1), \text{pasremove}(l_i, e_1)\}$.

- For $a_1 = \text{pasadd}(l_j, e_2)$, $A_2(s_t|a_1) = \{\text{actadd}(l_j, e_1)\}$.

- For $a_1 = \text{pasremove}(l_j, e_2)$, $A_2(s_t|a_1) = \{\text{actremove}(l_j, e_1)\}$.

- For $a_1 = \text{actadd}(l_j, e_2)$, $A_2(s_t|a_1) = A_2(s_t) - \bigcup_{i=1}^{m}\{\text{pasremove}(l_i, e_1)\} - \bigcup_{\substack{i=1 \\ i \neq j}}^{m}\{\text{pasadd}(l_i, e_1)\}$.

- For $a_1 = \text{actremove}(l_j, e_2)$, $A_2(s_t|a_1) = A_2(s_t) - \bigcup_{i=1}^{m}\{\text{pasadd}(l_i, e_1)\} - \bigcup_{\substack{i=1 \\ i \neq j}}^{m}\{\text{pasremove}(l_i, e_1)\}$.

$A_1(s_t|a_2) \subseteq A_1(s_t)$ can be similarly defined.

We can now redefine our transition function stochastically as follows:

$$P(s_t, a_1, s_{t+1}) = \sum_{\substack{a_2 \in A_2(s_t|a_1) s.t. \\ s_t \xrightarrow[<a_1, a_2>]{} s_{t+1}}} \frac{1}{|A_2(s_t|a_1)|}.$$

This can be readily generalized to some probability distribution over $A_2(s_t|a_1)$. Thus, this is the only modification needed.

In Chapter 7, e_2 will employ probabilistic transitions in terms of ϵ-greedy and τ-softmax as described in the background chapter.

Chapter 7

Modeling Opinion Formation in Dynamic Situations

7.1 Dynamic Task

In Chapter 6, we have assumed that we have complete knowledge of the environment and the decision-making style of the other player. Consider the following simple scenario: A used car salesman tries to sell a car by convincing customers that they should buy the car. Customers on the other hand may accept the salesman's argument or provide their own arguments to the salesman of why they should not buy the car (in hopes of getting a better deal). Over time, the salesman eventually learns a policy from experiencing many different customers. Clearly, the information requirements of the earlier formulations may not be met in this scenario. Here, we show how our approach can also account for this scenario.

Let us assume that the car salesman can get a sense of what cars a customer has already looked at, say from their opening conversation. This implies that the salesman knows the DTM for the customer and thus K_2^t. However, each customer is still likely to be quite different in terms of their individual goals which ultimately impacts the reward value and thus we are unlikely to have any information regarding transition probabilities.

Cast in terms of our framework, the salesman is aware of the DTMs, available actions, and reward values over time as well as across customers. This problem is classified as an online learning problem (Sutton and Barto, 1998). We can formulate it as follows: the state of the environment is defined by $s_t = (K_1^t, K_2^t)$. Agent e_1, the salesman, has the following goal:

$$\lim_{t \to +\infty} \{\gamma_1 |o_1^{t+1} - o_1^t| + \zeta_1 |o_1^{t+1} - o_2^{t+1}|\} = 0$$

where $\gamma_1, \zeta_1 \in [0, 1]$ are constants.

This alone is a model-free reinforcement learning model that requires on-line experience, i.e., learning as you go or learning on the job. Each customer represents an episode which is a finite sequence of time steps (called episodic tasks)(Sutton and Barto, 1998). Our car salesman problem can be solved using Q-learning for reinforcement learning. Recall from our description of Q-learning in Chapter 2, we can determine a good policy for the salesman by iterating over $Q(s_t, a)$ which is based on prior values of Q and the reward.

For our problem, we modified the algorithm more specifically as shown in Figure 7.1. We define the following:

- A is the action table where for each state s^t, $a_1 \in A(s^t)$ is the set of valid actions for e_1 such that there exists an action a_2 that is compatible with a_1 as defined earlier.

- Δ is a non-positive constant value assigned as the reward value for actions that are not valid for s_t.

- s^t is a terminal state if $|o_1^t - o_2^t| < \epsilon$ for some small positive ϵ.

Each episode is a customer and thus the selection criteria of e_2 (customer) will be the basis for determining the next state. Since each customer C_i is potentially different, the choice of customer action can be different for each episode which results in different state transitions and reward values. However, the Q-learning algorithm does not need to know the specific customer action selection (or even goals). For our car salesman, the reward function is some function based on the salesman's goal above.

By using Q-learning methods, we now can model dynamic opinion formation tasks as consisting of a sequence of (non-)episodic opinion formation tasks each of which may interact with different individuals.

7.2 Multi-agent Task

Up to now, we have provided solutions to model various forms of opinion formation tasks: episodic and non-episodic opinion formation tasks are concerned with whether the task is interactive between two entities; deterministic versus stochastic refers to varying degrees of uncertainties in the interactions between two entities; and, dynamic versus non-dynamic tasks concern with whether the task contains a sequence of atomic opinion formation tasks[1]. All the solutions we have provided so far considers two interacting agents.

What if multiple agents are involved in opinion exchanges? An opinion formation task among multiple agents can be cast into dynamic opinion formation tasks, each

[1]An atomic opinion formation task cannot be divided into multiple opinion formation tasks.

```
Input: States including terminal states
Input: Discount factor γ ∈ [0, 1]
Input: Maximum steps T some positive integer
Input: Action table A
Input: Step size α ∈ [0, 1]
Input: For C_1, C_2, ..., C_N, C_i is an episode (customer)
Input: Maximum steps T
Input: s_1 is the starting state for e_1 (salesman)
Output: Deterministic policy π such that π(s) = arg max_a Q[s][a]
var matrix Q[s][a] ← Δ if a ∉ A(s), 0 otherwise.
foreach episode (customer) C_i do
    var τ ← 1;
    var s_2 ← randomly selected starting state for e_2 (customer C_i);
    var s ← (s_1, s_2);
    while s is not terminal and τ <= T do
        Choose action a ← from A(s) using ϵ-greedy policy derived from Q;
        Take action a; observe reward r and next state s';
        Q(s, a) ← Q(s, a) + α[t + γ arg max_{a'∈A(s')} Q(s', a') − Q(s, a)];
        s ← s';
        τ ← τ + 1;
```

Figure 7.1: Q-learning Algorithm for Dynamic Opinion Formation Tasks

of which is an episodic task between two agents. This treatment is intuitive as typically one only listens to or talks to another individual due to cognitive limits in communication.

An alternative way is to define a multi-agent task considering opinion exchanges among them in one opinion formation task.

Definition 1 (Multi-agent MDP)**.** Let n be a positive integer > 1. Let $\mathcal{M} = \{M_1, M_2, \cdots, M_n\}$ be a collection of n MDPs where for each $i = 1, 2, \cdots, n$, $M_i = (S, A, P_i, R_i)$. Let $\alpha = \{\alpha_1, \alpha_2, \cdots, \alpha_n\}$ be an stochastic vector, i.e., $\sum_{i=1}^{n} \alpha_i = 1$. The joint MDP induced by $\mathcal{M}$ and α is the MDP $M_{\sigma_\alpha} = (S, A, P, R)$ where for each $s, s' \in S$ and $a \in A$, $P(s, a, s') = \sum_{i=1}^{n}(\alpha_i {*} P_i(s, a, s'))$ and $R(s, a, s') = \sum_{i=1}^{n}(\alpha_i {*} R_i(s, a, s'))$.

Let $\sigma = \{\sigma_1, \sigma_2, \cdots, \sigma_n\}$, where for each $i = 1, 2, \cdots, n$, σ_i is a policy of M_i. The joint policy σ_α with respect to σ and α is a randomized policy in the sense that at each time step t, the policy σ_i is selected with probability α_i, and applied to M_i. Thus, the value function of σ_α, for all $s \in S$, is

$$V_{\sigma_\alpha}(s) = \sum_{t=0}^{\infty} \gamma^t \sum_{s' \in S} \sum_{i=1}^{n} \alpha_i * P_i(s, \sigma_i, s') * R_i(s, \sigma_i, s')$$

for some $\gamma \in [0, 1]$.

Given stochastic vector α, σ_α is maximal with respect to α if and only if for all joint policy σ^*_α, $V_{\sigma_\alpha}(s) > V_{\sigma^*_\alpha}(s)$ for all $s \in S$.

Theorem 2 (Optimality for Multi-agent MDP)**.** Given a collection of $\mathcal{M}$ of MDPs, and a corresponding collection of policy σ. For any stochastic vector α, σ_α is maximal if and only if for each i, σ_i is an optimal policy of M_i.

Proof: The result follows from the Definition of $V_{\sigma_\alpha}(s)$.

The above Theorem provides a method for determining the maximal randomized policy for multiple MDPs by determining the optimal policy of each individual MDP.

If the S's and/or the A's are not the same in the MDP's, we can always take the union of all such Ss and the union of all such As and define the rest accordingly. Thus, there is no loss of generality by assuming that all the Ss are equal, as well as, all the As are equal.

Moreover, if the reward function is the same for all the MDPs, then the reward function for $V_{\sigma_\alpha}(s)$ is equal to the common reward function. This is true regardless of the αs.

7.3 Case Study

7.3.1 A Case Study on Training Design

We study the following hypothetical problem:

> *Problem Setting*: We want to train advocates at Dartmouth College that are proficient at convincing others to believe it is a great university. We have materials about different universities, but unfortunately we cannot recruit too many people with a wide variety of backgrounds (personality, holding different opinions, preferences in communication) to practice with.
> *Target Problems*: What types of training are critical to provide? What suggestions (strategies) can we provide the advocates to increase their skills in debating immediately?

In Chapter 6, we introduced goal functions with two control parameters ζ and γ. ζ can be intepreted as the passivity-activism scale from 0 to 1 representing an agent's eargerness for reaching a consensus and γ can be intepreted as malleability-idealism scale from 0 to 1 representing an agent's willingness to change its own opinion. In the experiments, we focus on situations where consensus is actively sought for but individuals differ in the way to reach consensus. Thus in this case study, we focus on two goal profiles: IA profile ($\gamma = 1$ and $\zeta = 1$) and MA profile ($\gamma = 0$ and $\zeta = 1$).

7.3.2 Dataset Construction

We designed a synthetic dataset in the following manner: First, we collected and pre-processed the U.S. News 2013 College Data[2]. The College data (in Table 7.1) contains seven attributes of five universities. Table 7.2 shows the five universities we

[2]Data is collected from `http://colleges.usnews.rankingsandreviews.com/best-colleges/rankings/national-universities`

have chosen (randomly) for this case study along with the discretized feature vectors.

Table 7.1: U.S. News 2013 College Data

Feature	Value#0	Criteria#0	Value#1	Criteria#1
Ranking	0	< 100	1	≥ 100
Tuition total enrollment	0	$< \$25,000$	1	$\geq \$25,000$
Fall 2011 acceptance rate	0	$< 35.00\%$	1	$\geq 35.00\%$
Average freshman retention rate	0	$< 86.00\%$	1	$\geq 86.00\%$
6-year graduation rate	0	$< 71.05\%$	1	$\geq 71.05\%$
Classes with under 20 students	0	$< 47.00\%$	1	$\geq 47.00\%$
SAT/ACT 25th-75th percentile	0	$< 1,010$	1	$\geq 1,010$

Table 7.2: Feature Vectors for Five Universities

University	Feature Vector
University of California-Los Angeles	0 0 1 0 1 1 1 1 1
Columbia University	0 1 1 0 1 1 1 1 1
Georgia Institute of Technology	0 0 1 1 1 1 0 1 1
Stevens Institute of Technology	1 1 0 1 1 1 0 1 1
Worcester Polytechnic Institute	1 1 0 1 1 1 1 1 1

Next, we simulated trainees, trainers and testers. Trainees refer to the advocates at Dartmouth College, and trainers refer to individuals that are recruited to practice with the trainees. Testers are the individuals who trainees must aim to convince them that a Dartmouth College is a tier 1 university. We can imagine trainers and testers to be prospective or current students, family members of students, partners and even competitors.

We consider two goal profiles for each trainee: either MA-style (malleability and active) or IA-style (idealist and active). A MA-style trainee is a listener, he would be willing to adjust his opinions in order to reach a conesensus. An IA-style trainee is aggressive, he does not adjust his opinion but seeks ways to convince the other one to reach a consensus.

We consider a wide range of trainers and testers to cover individuals with different backgrounds. We generated 18 different types of trainers and testers who differ in both goal profiles (See Table 7.3) and their behavioral styles. We simulate three behavior styles: $pureGreedy$, $\epsilon - greedy(0.05)$, and $\tau - softmax(1.0)$.

Table 7.3: Goal Profiles for Trainers and Testers

Goal Profile	Parameters
pureIA	$\gamma = 1.0$ and $\zeta = 1.0$
pureMA	$\gamma = 0.0$ and $\zeta = 1.0$
pureMP	$\gamma = 0.0$ and $\zeta = 0.0$
pureIP	$\gamma = 1.0$ and $\zeta = 0.0$
mixedImixedP (mImP)	$\gamma = 0.75$ and $\zeta = 0.25$
mixedMmixedA (mMmA)	$\gamma = 0.25$ and $\zeta = 0.75$

To complete our design for this dataset, we consider the following queries shown in Table 7.4. The feature with value x is the target for the opinion, which in this case is the ranking of a university. The feature with ? represents a missing value.

Table 7.4: Feature Vector for Two Queries

Query	Feature Vector
q_1	x 1 ? ? ? ? ? ? ? 1
q_2	x ? ? ? ? ? 1 ? 1

7.3.3 Simulations

We simulate opinion formation tasks a trainee may engage in regarding all possible trainers and testers. Every simulated trainee has a knowledge basis for his initial opinion (one out of 32 possible combinations of university vectors), a goal profile (MA or IA). As we use Q-learning methods in this case study, all the trainees have the same behavior style $\epsilon - greedy(0.05)$. Every simulated trainer and tester has a knowledge basis for his intial opinion (one out of 32), a different goal profile (one out of six shown in Table 7.3), and a different behavior style (one out of three).

We consider two forms of dynamic opinion formation tasks - tasks where a trainee interacts with the same type of trainer (or a tester) and tasks where a trainee interacts with trainers (or testers) of different types. Every dynamic opinion formation task (we refer it as one behavioral run) has 1000 episodic opinion formation tasks. Within a dynamic opinion formation task, the trainee keeps changing its opinion by interacting with trainers (or testers) in a sequential manner. If a pair of individuals do not reach consensus, we terminate their discussions after ten rounds of interactions and the trainee then continues to the next one in the queue. The initial opinion is randomly generated for the trainer (or a tester) in each opinion formation task. Within one behavioral run, a trainee may talk to agents of different types. For example, a trainee talks to pureIA-style agents for 500 times and talk to pureIP-style agents for 500 times. In particular, for our testbed, the nine composite behavioral runs are as follows (these were randomly generated):

#2 (pureIA, ϵ-greedy) , (mMmA, ϵ-greedy)

#2 (pureIP, τ-softmax), (pureMA, ϵ-greedy)

#2 (mMmA, pure-greedy), (pureIA, ϵ-greedy)

#2 (pureMP, ϵ-greedy), (mImP, pure-greedy)

#2 (pureMP, pure-greedy), (pureMA, ϵ-greedy)

#5 (pureIA, ϵ-greedy), (mMmA, ϵ-greedy), (pureMP, pure-greedy), (pureMA, ϵ-greedy), (pureIP, τ-softmax)

#5 (pureMa, ϵ-greedy), (pureMP, ϵ-greedy), (mImP, pure-greedy), (mMmA, pure-greedy), (pureIA, ϵ-greedy)

#10 (pureIA, ϵ-greedy), (mMmA, ϵ-greedy), (pureMP, pure-greedy), (pureMA, ϵ-greedy), (pureIP, τ-softmax), (pureMA, ϵ-greedy), (pureMP, ϵ-greedy), (mImP, pure-greedy), (mMmA, pure-greedy), (pure-IA, ϵ-greedy)

#10 (mImP, τ-softmax), (mImP, ϵ-greedy), (mImP, τ-softmax), (pureMA, ϵ-greedy), (pureIP, τ-softmax), (pureIP, ϵ-greedy), (pureMP, pure-greedy), (mMmA, ϵ-greedy), (pureMP, ϵ-greedy)

We conducted simulations for both all pairs of trainee-trainer and pairs of trainee-tester. In total, we have around $140,000$ behavioral runs completed. Each behavioral run represents one dynamic opinion formation task consists of 1000 episodic opinion formation tasks. The parameters in our specific Q-learning algorithm are as follows:

Table 7.5: Parameters for Q-learning Methods

Parameter	Value
Discount Factor	0.5
Step Size	0.1
Epsilon Termination	0.005
Δ Illegal Action	-100
ϵ-greedy value	0.05
Max Turns	10
# of behavior runs	1000

7.3.4 Results

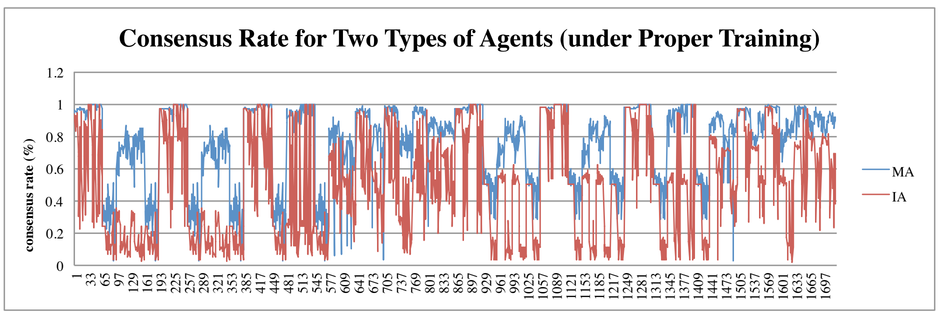
To facilitate our discussion we introduce two situations during training: proper training and improper training. Proper training refers to situations where the training experience a trainee receives matches the problems he encounters with testers. In other words, the type of trainers a trainee interacts within a training run is the same as the type of testers a trainee interacts within a testing run. In the situation with composite behavioral runs, it is considered proper training if the composite is the same for both training and testing cases. On the other hand, improper training refers to situations where the training experience a trainee receives does not match the problems he encounters later with testers.

Improper training is a very typical situation we face in the real world: providing sufficient coverage of training is labor expensive and time-consuming. A doctor in a hospital gains more experience on how to convince patients or how to effectively reach a consensus by practicing medicine over time. Similarly, a car salesman gains experience on how to sell cars by engaging with a variety of potential buyers every day. This framework can be used to simulate different problems and thus provide suggestions and analyses for possible outcomes. If budgets are available for providing training, this framework can be used to simulate different cases in order to help planning the training process. The framework can also be used to help in making hiring decisions.

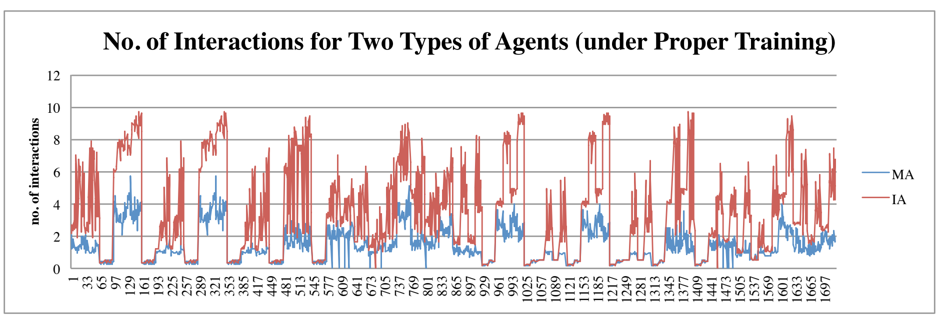
This section will answer various questions centering on the topic of training design from the simulation results. The purpose of these results is to demonstrate the usefulness of this framework, to illustrate that the opinions are computable and learnable, and to show its compliance with social theories and theories of rationality.

Question 1: What type of trainee can perform better under proper training?

To compare performances among different types of trainees, we introduce *consensus rate* and *number of turns* to measure success at reaching consensus in each behavioral run. We compute consensus rate as the percentage of opinion formation tasks that result in consensus out of 1000 tasks in total. We compute an averaged number of turns to measure the speed of consensus within each behavioral run. As an opinion formation task terminates after 10 rounds of interactions, the upperbound of an averaged number of turns is 10. Figure 7.2 compares these two performance metrics for trainees with different goal profiles. Trainees with MA-style ($\mu_1 = 0.775$, $\sigma_1 = 0.058$ for consensus rate, $\mu_2 = 1.428$, $\sigma_2 = 1.058$ for no. of turns) significantly outperform IA-style ($\mu_1 = 0.503$, $\sigma_1 = 0.098$ for consensus rate, $\mu_2 = 3.441$, $\sigma_2 = 7.742$ for no. of turns) for both performance metrics ($p_1 = 2.75E - 162$ for consensus rate and $p_2 = 1.842E - 157$ for no. of turns).



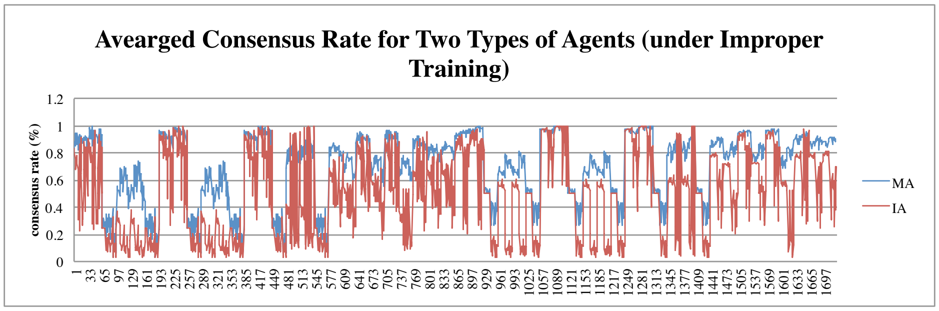
(a) Comparisons of Consensus Rates between MA-style and IA-style Agents under Proper Training



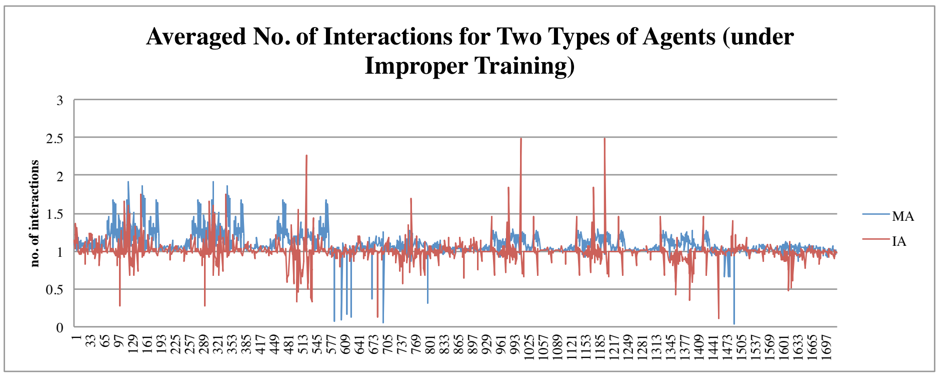
(b) Comprisons of Number of Interactions Taken to Reach Consensus between MA-style and IA-style agents under Proper Training

Figure 7.2: Performance Comparisons between MA-style and IA-style Agents under Proper Training

Question 2: What type of trainee can perform better under improper training?



(a) Comparisons of Consensus Rates between MA-style and IA-style Agents under Improper Training



(b) Comprisons of Number of Interactions Taken to Reach Consensus between MA-style and IA-style Agents under Improper Training

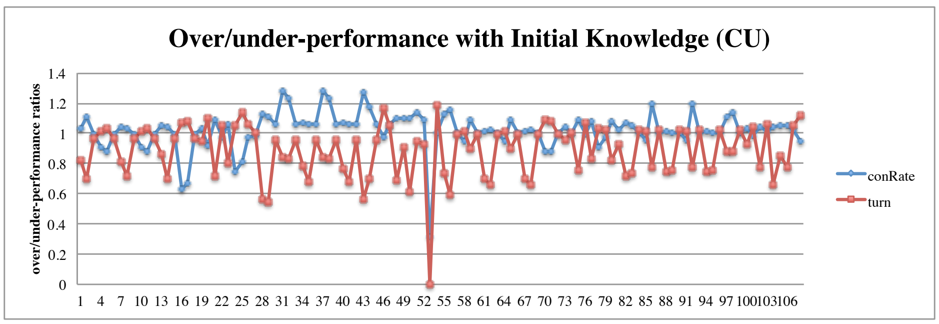
Figure 7.3: Performance Comparisons between MA-style and IA-style Agents under Improper Training

Even though trainees with MA-style perform better than IA-style trainees when proper training was received, the results shown in Figure 7.3 tell a slightly different story. IA-style trainees on average ($\mu = 0.997$, $\sigma = 0.019$ for no. of turns) significantly take fewer turns to reach consensus compared to MA-style trainees ($\mu = 1.098$, $\sigma = 0.022$ for no. of turns with $p = 6.40 - E92$) [3]. Despite the fact that MA-style trainees perform well when the type of training they receive matches well with the testing situation, IA-style trainees perform better at convincing unexpected types of testers. The results suggest that the performance we observe in training are not representative of how they might actually perform. What can we conclude from the simulation results for question 1 and question 2? First of all, if sufficient

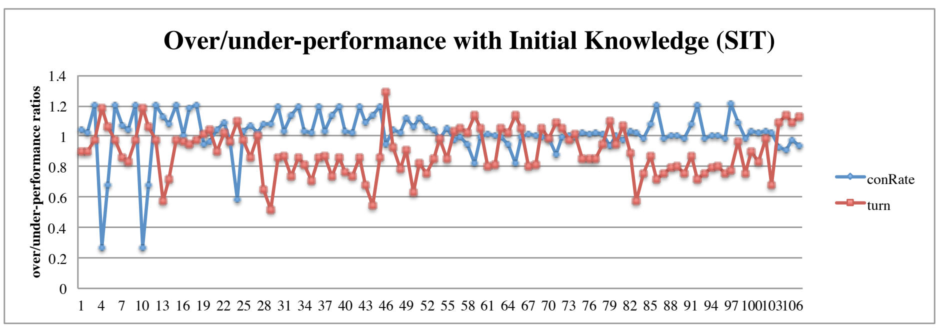
[3]The performance under improper training is averaged thus on a different scale compared to plots in Figure 7.2.

training can be provided, MA-style trainees are preferred as they achieve better performances compared to IA-style trainees. On the other hand, IA-style trainees are preferred if sufficient training cannot be provided. These comparisons also indicate the importance of proper training.

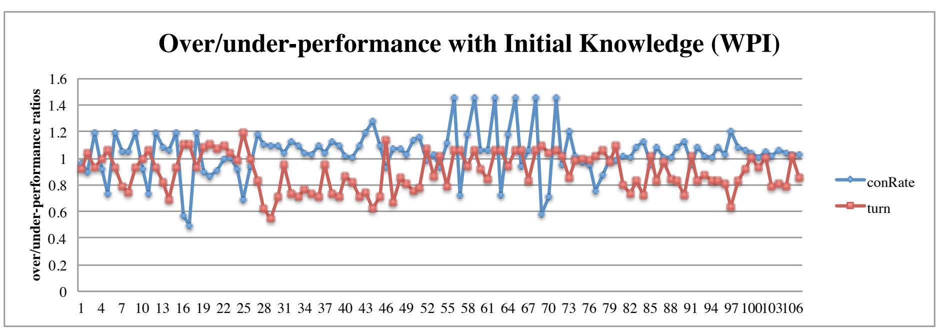
Question 3: Does knowledge matter in handling unexpected testers?



(a) Over/under-performance with Initial Knowledge about Columbia University

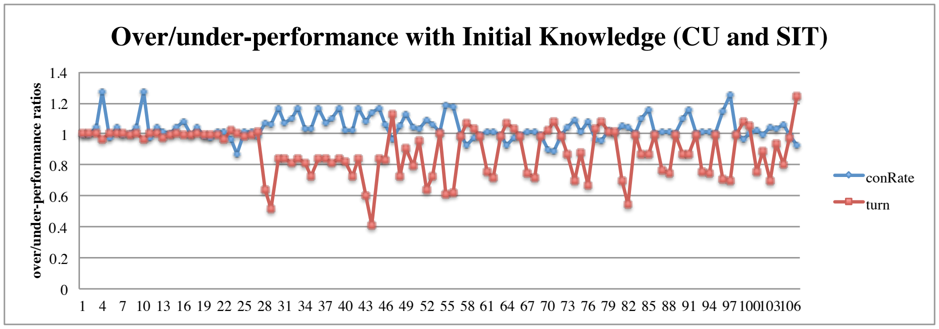


(b) Over/under-performance with Initial Knowledge about Stevens Institute of Technology

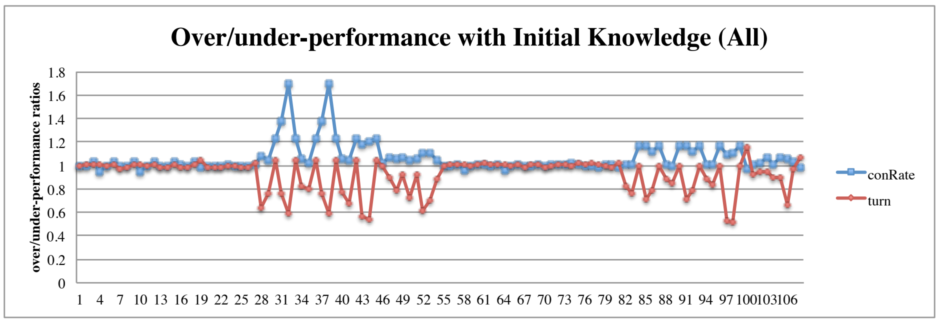


(c) Over/Under-performance with Initial Knowledge about Worcester Polytechnic Institute

Figure 7.4: Over/under-performance with Different Initial Knowledge (Part A)



(a) Over/under-performance with Initial Knowledge about Columbia University and Stevens Institute of Technology



(b) Over/under-performance with Initial Knowledge about All Universities

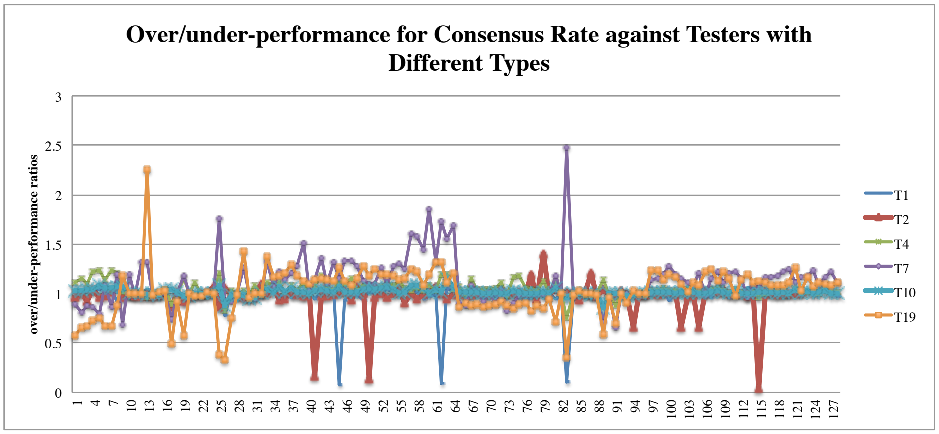
Figure 7.5: Over/under-performance with Different Initial Knowledge (Part B)

Next, we wish to explore whether relying on different knowledge would help a trainee performs better when encountering an unexpected tester. We compute over/under-performance metric for both the consensus rate and number of turns to reach a consensus by dividing the averaged measure for all situations of improper training by the measure for situation of proper training. An over/under performance metric of value 1.0 indicates that performance under proper and improper training is the same. Figure 7.4 and Figure 7.5 show results for five different knowledge setups. If a trainee only considers one learning episode when engaging in a task with a tester, Figures 7.4a, 7.4b, and 7.4c all show some evidences that trainees reach consensus faster when encountering a new type of tester. However on the other hand, trainees take more turns to reach a consensus with a new tester. In fact, the opinion derived based on knowledge about Stevens Instuitute of Technology and the opinion derived based on knowledge about Columbia University is the same (both with emitted probabilities 1.0). According to the plots (Figure 7.4b and Figure 7.4c), the performance metrics are different for these two situations. This shows that our framework is compliant with theories in rationality that the preferences of cognitive states should differ regardless of whether they map to the same emitting probabilities or not. These results also validate that our framework differs from linear models which simply aggregate positive/negative evidences.

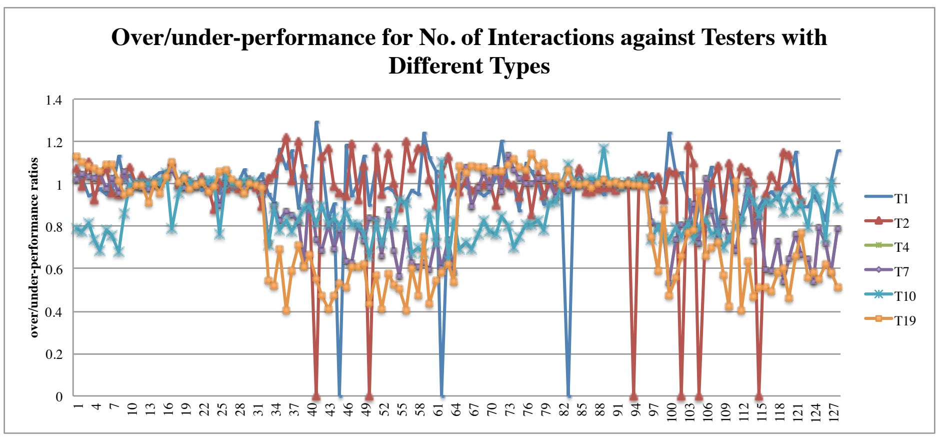
Furthermore, as shown in Figure 7.6b and Figure 7.5b, an agent with more knowledge in its initial working memory performs consistently no matter whether proper training is received. The dynamics of over/under-performance for different degree of knowledge invovlement supports Gigenrenzer's hypothesis (Gigerenzer and Brighton, 2009) that biased minds are often observed to make better inferences. Relying on less knowledge helps human remain flexible in coping with new problems in various situations. A natural extension from this is to test whether humans would develop different patterns if located in different environments - simplier cognitive states are preferred under uncertain environments and more complicated cognitive states are preferred under certain environment.

Question 4: What type of testers are in generally harder to reach a consensus with?

Will some people be naturally harder to deal with? Figure 7.6 shows the performances of a trainee when engaged in tasks with six different testers (detailed descriptions on the types are shown in the list below). T19-type and T7-type testers are hardest and T2-type testers is the easiest. Both the composite behavioral runs are easier for trainees to reach a consensus indicating that it is more beneficial to train an individual with a more diverse set of trainers. Among the four singleton



(a) Over/under-performance for Consensus Rates against Testers with Different Types

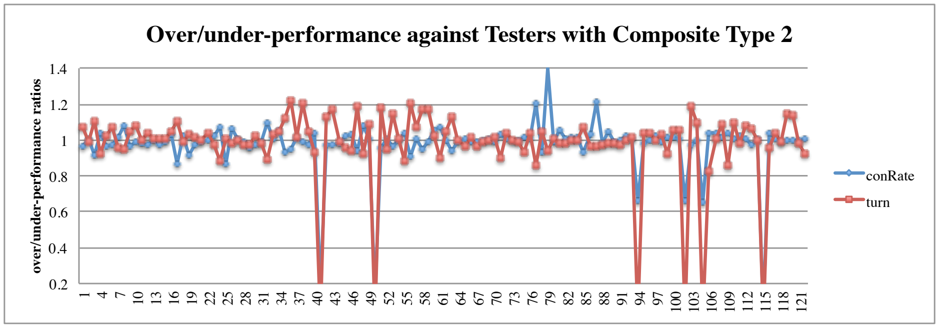


(b) Over/under-performance for Number of Interactions against Testers with Different Types

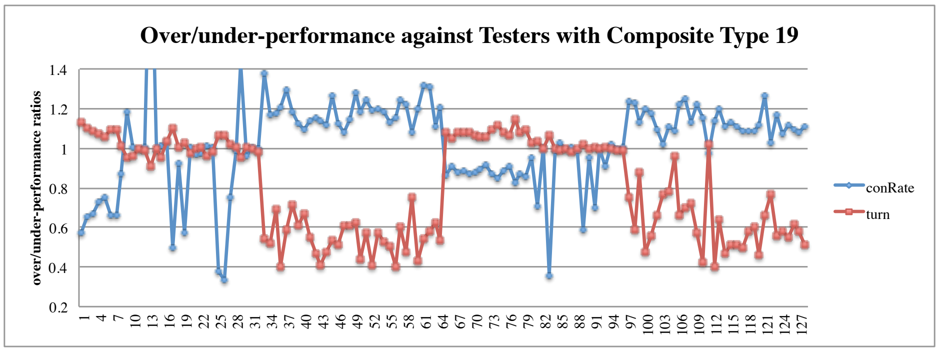
Figure 7.6: Over/under-performance against Testers with Different Types

types of behavioral runs, the passive testers (T19 and T7) are harder than the active testers (T4 and T7). This is intuitive because the active testers are more cognitively engaged than the passive testers because they also desire to reach consensus with the trainee. The performances varies a lot for each type of tester. The detailed performance metrics for T2 (easiest) and T19 (most difficult) are shown in Figure 7.7.

- T1: s+10+-IA:eG=0.05=-mMmA:eG=0.05=-MP:PG-MA:eG=0.05=-IP:tSM=1.00=-MA:eG=0.05=-MP:eG=0.05=-mImP:PG-mMmA:PG-IA:eG=0.05=-1000



(a) Over/under-performance for Consensus Rates against Testers with Different Types (Hardest Type)



(b) Over/under-performance for Number of Interactions against Testers with Different Types (Easiest Type)

Figure 7.7: Easiest and Most Difficult Types of Testers

- T2: s+10+-mImP:tSM=1.00=-mImP:eG=0.05=-mImP:tSM=1.00=-MA:eG=0.05=-IP:tSM=1.00=-IP:eG=0.05=-MA:PG-IA:PG-mMmA:eG=0.05=-MP:eG=0.05=-1000
- T4: s+1+-IA:PG-1000
- T7: s+1+-IP:PG-1000
- T10: s+1+-MA:PG-1000
- T19: s+1+-MP:PG-1000

7.3.5 Relations with AI

This thesis sits at the heart of Artificial Intelligence and Cognitive Science, the area that the co-founder of AI, Marvin Minsky, considers to be dead since the 1970s:

> "AI has been brain-dead since the 1970s," said AI guru Marvin Minsky in a recent speech at Boston University. Minsky co-founded the MIT Artificial Intelligence Laboratory in 1959 with John McCarthy.[4]

Advances in areas on human cognitive behavior have been slow for the last few decades. For instance, the cognitive architecture SOAR has been an ongoing effort for decades. There has been tremendous progress made in learning, vision, robotics system, and even speech recognition, but unfortunately the successes in these problems have not helped much in developing a better AI system that "thinks like humans".

Despite how challenging this task is, we still believe it is hopeful and also a critical step to overcome in order to make machine systems truly intelligent. The systems that simulate human intelligence can in turn help almost every field that relates to human intent and human behavior - they can help economists to simulate behavior and decisions that they can then test various theoretical accounts; they can help social scientists to encapsulate social behavior to save time and effort in conducting large-scale empirical studies - ultimately, they can help the entire society to be predictive of human intent and be normative for human behavior.

This thesis presents a *new research methodology* that may be useful in advancing the field: transcribing human behavior (actions) into human thinking (cognition). We consider a handful of actions such as communicate an opinion, select supporting/non-

[4]This story ran on page D1 of the Boston Globe on 5/26/2003. Its content can be found at `http://web.media.mit.edu/~lieber/Press/Globe-Common-Sense.html`

supporting arguments to send, and determine whether to accept or reject receiving messages. Indeed, the actions we attempt to transcribe have a wide spectrum for activities involving opinions. As actions presented in activities about opinion is knowledge-centric and these actions are decisions requiring a variety of cognitive processes (i.e., generating an argument is achieved by reasoning, selecting an argument to send is a decision-making problem itself, and determining whether to accept or reject an argument has some relation to strategy). To fully model these actions in order to make them computable and tractable, the only evident way is to develop a cognitive model that is "intelligent" at perferming these actions.

In order to transcribe actions into cognition, we carefully analyze empirical conclusions within both topics and then to find the right mappings to transcribe. In this thesis, the mappings we find between human thinking (opinion reasoning) and human behavior (opinion exchanges) for this work is the *learning nature of opinion formation and decision nature of opinion change* which is achievable by the existence of powerful computational methods such as reinforcement learning. Then, to capture the semantics of behavior into a cognitive model, we need to carefully evaluate theories in rationality - not what types of reasoning occur but essentially what types of reasoning the existing computational methods have failed at. With that, we can make sure the design of the cognitive model is not biasd by the need to match actual human behavior and that the cognitive model has maintained the same semantics between human behavior and human cognition. The more methodical the approach is, the less artifacts we may have in the final system.

The effort presented in this thesis has benefited from many areas:

- We rely on theories of bounded rationality in the design of a DTM (diversities in human reasoning and insufficiencies in human reasoning)

- We benefited from empirical findings on smart human heuristic reasoning to partially evaluate a DTM
- We rely on conclusions from psychology and neuroscience in the design of a DTM (working-memory, long-term memory, questions etc.)
- We rely on progress in computer science for problem solving and modeling (Markov decision process, reinforcement learning methods)
- We rely on social theories to understand the problem itself and to discover the parellel relationships we can possibly establish between human behavior and human thinking (numerous theories on opinions such as RAS, CII, and ACT)

This is essentially why we believe that overcoming the problem of AI has to be a large-scale multi-disciplinary effort: axioms and theories in the field sociology, psychology etc. can shed light on behavior mechanisms and thinking mechanisms, the field of mathematics can help define the framework so that its construction is more accessible and the underlying assumptions are easier to evaluate, the field of computer science can help provide tools and algorithms in problem solving and analyses, and lastly the multiple fields together can evaluate or at least demonstrate the usefulness of the framework when applying it to study problems across various domains.

As the overall objective of this thesis is to develop a general framework of computational opinions that unifies various dominant social theories and theories of rationality, it is also hard to evaluate this framework. One possible way to evaluate this framework is via the famous turing test proposed by Alan Turing in his 1950 paper (Turing, 1950). The Turing test evaluates the intelligence level of a system by testing whether its behavior is indistinguishable from an actual human. For example, a human judge engages in a natural language conversation with either a human or a machine and each judge hypothesizes whether he/she thinks the conversations is conducted with a real human (Weizenbaum, 1966). However, it has been widely criticized that a computer system that can mimick human behavior well can pass the test without needing to be capable of thinking (Saygin et al., 2003). The evaluation for this framework may need to be conducted by researchers from various domains by testing their hypotheses or through simulations where ground truths are available.

Chapter 8

Conclusion

We have developed a framework for computational opinions that models sequences of opinion formation and opinion change tasks. We formulated our framework by recognizing the learning nature of opinion change and the decision-making nature of opinion formation. This perspective enabled us to clearly identify the impacts of external influents and how they are internalized interleaved with how internal influents arise and externalized as opinions. Our underlying cognitive model, the Double Transition Model (DTM), firstly provides the mechanism for influence and how influence transforms our opinions both in terms of our working knowledge and specific query form. Secondly, as DTMs were not meant to account for the causes of such influences, we converted the undirected edges in DTMs into directed edges representing various forms of influence. Then, we were able to formulate opinion formation tasks with external influence as Markov decision problems. We constructed MDPs based on DTMs and goal functions to capture the decision-making nature of opinion formation. Lastly, our framework is compatible with prevalent social theories of opinion change and theories of bounded rationality as well as being able to formally model known human heuristic reasoning. To the best of our knowledge, our results provide the first mathematical and computational framework that has been successful in accounting for these theories and heuristics. In essence, we have been able to provide a unifying yet general framework for defining and analyzing the wide variety of opinion change and formation. In the rest of this chapter, we will describe our contributions in detail and discuss future (and far term) work.

8.1 A Framework of Computational Opinions

The overall objective of this thesis is to model a sequence of opinion formation tasks. In this section we list the opinion formation tasks each chapter attempts to tackle to provide an overview of our solutions as well as requirements needed in modeling each task.

1. A sequence of (non-)dynamic (non-)episodic opinion formation tasks. (Effort in Chapter 4-5)

    It is extremely challenging to solve this most general problem. We try to tackle it by designing a cognitive model to achieve a sufficient coverage of possible states from which an opinion can be derived.

2. One non-episodic opinion formation task with no search. (Effort in Chapter 4-5)

    This is the most restricted form of a task. It has been addressed by modeling a cognitive state with a paired working-memory and a query on an issue. For this task, an opinion is derived from a full working-memory (equivalent to long-term memory) and a full specification of a query.

3. One episodic opinion formation task with complete knowledge of another agent. (Effort in Chapter 6)

This task starts with a known state, and the preferences of cognitive states are learned via solving a Markov decision problem with the other entity. The formation task is episodic since it consists of multiple iterations of message passing (including both arguments and opinions). Solving this MDP problem has been theoretically proven to converge to an optimal value under the assumption of full knowledge of the other entity's DTM. (see Figure 8.1a)

4. One non-episodic/episodic opinion formation task with complete knowledge of another agent. (Effort in Chapter 6)

    It is straight-forward to generalize the solution from last bullet by setting the horizon in a MDP to 1.

5. One episodic opinion formation task with training data. (Effort in Chapter 7)

    This task starts with a known state, but the preferences of cognitive states are learned via solving a Makov decision problem with the other entity using Q-learning. This frees us from the assumption of having complete knowledge of the other entity using Q-learning to approximate optimal policy. However, the Q-learning methods need sufficient numbers of training to reach a reasonble performance. This assumption is different from the assumption of full knowledge of the other entity's DTM. It is also intuitive since in real-world problems human transition from being novices to experts through repetitive learning/training practices. (see Figure 8.2a)

6. One (non-)episodic opinion formation task with training data. (Effort in Chapter 7)

It is straight-forward to generalize the solution from the last bullet by setting the allowed number of turns to 1.

7. One multi-agent opinion formation task. (Effort in Chapter 7)

    There are two approaches to model a multi-agent opinion formation task: one approach is to cast a multi-agent task into a dynamic opinion formation task, an alternative approach is to compute optimal actions a primary agent should take by solving a multi-agent MDP problem. We provide formal definitions in Chapter 7 and also proved that an optimal policy derived for the entire group is also optimal for each individual in that group.

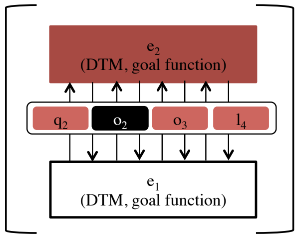
8. A sequence of (non-)dynamic (non-)episodic opinion formation tasks. (Effort in Chapter 7)

    The framework can now model a sequence of opinion formation tasks by treating them as a sequence of decision problems. For each decision problem, depending upon which assumption holds, we can select the appropriate reinforcement learning methods. In particular, if a reasonable understanding of the other entity is avaialble, then we can solve a MDP via exact methods such as value-iteration or policy-iteration algorithms. On the other hand, if observations of one's practice is available, then we can approximate optimal solutions via Q-learning methods. (see Figure 8.2b)

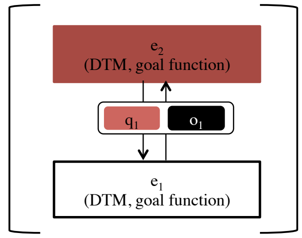
8.2 Future Work

1. Dynamic DTM

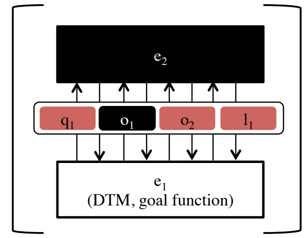
    Due to the complexity of the problem, we simplify the problem by assuming



(a) Episodic Opinion Formation Task with Complete Knowledge of e_2

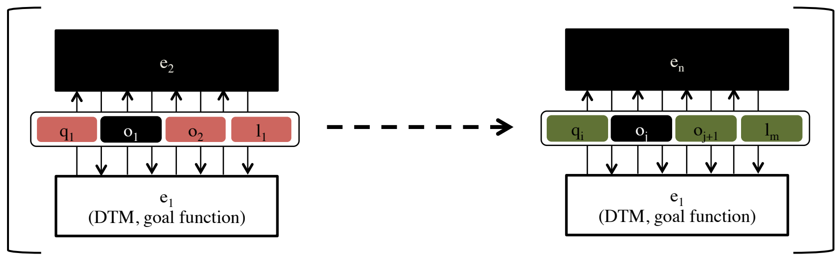


(b) Non-Episodic Opinion Formation Task with Complete Knowledge of e_2

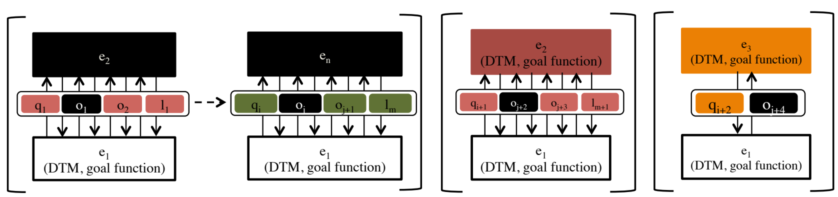


(c) Episodic Opinion Formation Task with Incomplete Knowledge of e_2

Figure 8.1: Three Forms of Opinion Formation Tasks



(a) Dynamic Episodic Opinion Formation Task with (In)complete Knowledge of the other agent



(b) A Sequence of Opinion Formation Tasks

Figure 8.2: A Sequence of Opinion Formation Tasks

a static and complete DTM. A DTM is static as it does not conduct online (real-time) learning when new observations are available. In the future, we aim to extend the framework to also include pure learning tasks (obtaining new knowledge by observations). Thus, learning of a DTM becomes online-learning versus offline-learning. It will also improve our simulation of one individual's state space and the pure learning tasks can be interleaved with opinion formation tasks. This is straight-forward to generalize as each task is independently solved based on the DTM at that time. It has not yet been implemented but simply needs an algorithm to update a DTM.

2. Different types of perceptions

   Current perceptions we consider are knowledge-centric as they have direct impact on the structure and the content in a DTM. In the future, we aim to extend the framework to also include the perception of actions (e.g. body language from another entity). By doing so, it can significantly increase the coverage of the problems as the forms of external influence is extended.

3. Incomplete DTM

   In this thesis, we assume that a DTM is complete; that is, a DTM is a complete graph and thus every pair of cognitive states is connected. As shown in Figure 1.2, temporal relations can be one of the principles we rely on to remove/add connections in the cognitive space. At the end of opinion formation task, the utilities (values) on a state can also provide useful insights on how to modify connections in the cognitive space. This idea is compatible with the idea of the spreading activation mechanism in a brain (Warren, 1977). Furtheremore, this is naturally one part of what has been learned through internalizing external influences. By better constraining the connections in a DTM, the cognitive model can function more like a human brain. In addition, a DTM with fewer connections will also dramatically decrease the size of the states in a Markov decision problem. The DTM itself can be used by researchers in fields of neuroscience and psychology to testify their theoretical accounts of how working memory is formed and how knowledge gets recalled.

4. Multi-Agent Opinion Formation Tasks

   We have provided definitions of multi-agent MDPs to model multi-agent opinion formation tasks. As there is a large body of MDP works on multi-agent MDPs considering both competitive and non-competitive scenarios, a natural extension of this framework is to define multi-agent MDPs based on the DTMs of multiple agents for various scenarios. We can leverage some work in the existing literature to incorporate game-theoretic multi-agent MDPs (Littman, 1994; Shoham et al., 2003), cooperative MDPs (Xuan et al., 2001) and competitive MDPs (Graepel et al., 2004) into our framework. This future work will significantly increase the coverage of the problems we can study.

5. Identification of Opinion Formation Tasks

   In this thesis, we define an opinion formation task to be a task on the same issue in the same context. The meaning of context is still vague as it can imply properties of a task in terms of time (whether a task happends in a consecutive sequence over a reasonable time), in terms of the communicators who are involved (two or multiple agents talking to each other) and so forth. Our current framework identifies a task soely based on the window of time and the identities of agents involved. However, if an opinion formation task does not happen as a sequence of continous activities, we may have the issue of persistence of opinion change which violates the assumption we made in a decision-making problem to solve episodic opinion formation tasks. We can search from the current empirical observations and establish a reasonable hypothesis of the conditions under which the persistence holds. Furtheremore, we wish to explicitly model a control parameter into the framework to decide the start and the end of an opinion formation task.

8.3 Future Future Work

1. Estimation of DTM

   By completing the future work, a framework can then build dynamic DTMs, handle different types of perceptions, use a more principled way to maintain connections between working-memories, and handle multiple-agent opinion formation tasks. The framework will be quite powerful in tracking the entire sequence of human behavior and perception. Multiple types of computations on opinion are already available with the existing framwork which include the functions to: 1) decide the best action to take within episodic opinion formation tasks, 2) process incoming opinions from others, and 3) derive opinions.

One piece that is missing but will be beneficial to the study is that the initial DTM as a representation of the entire behavior and perception of an individual may not be completely available. One approach to estimate the initial DTM is to leverage our existing conclusions from social networks to: 1) estimate an individual's DTM based on another individual, 2) estimate the changes in one's DTM based on changes we have tracked for another individual if they have engaged in "hidden" communications with a third individual.

For example, we can estimate a supervisor's DTM as a merged product of his/her employees' DTMs. If there is minimal communication between a supervisor and his/her employees, we can estimate the supervisor's DTM as a subset (or a compression) of a merged product of his/her employee's DTMs. Of course, we can inversely learn how an enterprise culture may look like given the relationship between a supervisor's DTM and those of his/her employees.

The second approach will benefit the problems such as predicting individuals' opinions and behavior when only the individuals he/she communicates with are visible.

2. Inverse Reinforcement Learning

   The entire thesis describes how to model/instantiate such a framework for computational/predictive purposes. However, we can also utilized this framework to conduct inverse learning (Ng and Russell, 2000). In the current framework, the parameters of goal fuctions are inputs to the framwork in order to capture the intent of individuals. In the future, we can learn/tune the parameters to be in line with the opinion dynamics being observed. The final paremeters that are learned can now be useful to characterize individuals - their intent and goals, which can benefit predictive analyses.

Here, we complete our presentation on the framework of computational opinions.

Bibliography

Mohammad Afshar and Masoud Asadpour. Opinion formation by informed agents. *Journal of Artificial Societies and Social Simulation*, 13(4):5, 2010.

Dana Angluin and Carl H Smith. Inductive Inference: Theory and Methods. *ACM Computing Surveys (CSUR)*, 1983.

Nikolay Archak, Anindya Ghose, and Panagiotis G Ipeirotis. Show me the money!: deriving the pricing power of product features by mining consumer reviews. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 56–65. ACM, 2007.

Elliot Aronson, Judith A Turner, and J Merrill Carlsmith. Communicator credibility and communication discrepancy as determinants of opinion change. *The Journal of Abnormal and Social Psychology*, 67(1):31–36, 1963.

W Brian Arthur. Inductive reasoning and bounded rationality. *The American economic review*, 84(2):406–411, 1994.

Alan D Baddeley and Graham Hitch. Working memory. *The Psychology of Learning and Motivation*, 8:47–89, 1974.

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern Information Retrieval*. ACM press New York, 1999.

Lotte Bailyn and Herbert C Kelman. The effects of a year's experience in america on the self-image of scandinavians: A preliminary analysis of reactions to a new environment. *Journal of Social Issues*, 18(1):30–40, 1962.

Ray Bareiss, Bruce W Porter, and Kenneth S Murray. Supporting start-to-finish development of knowledge bases. In *Knowledge Acquisition: Selected Research and Commentary*, pages 13–37. Springer, 1990.

Robert Baron, S. So right it's wrong: Groupthink and the ubiquitous nature of polarized group decision making. *Advances in Experimental Social Psychology*, 37: 219–253, 2005.

Richard Bellman. A problem in the sequential design of experiments. *Sankhyā: The Indian Journal of Statistics (1933-1960)*, 16(3/4):221–229, 1956.

Bernard R Berelson, Paul F Lazarsfeld, and William N McPhee. *Voting: A study of opinion formation in a presidential campaign*. University of Chicago Press, 1986.

Joseph Berger and Morris Zelditch. *New directions in contemporary sociological theory*. Rowman & Littlefield, 2002.

Steven Bethard, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. Automatic extraction of opinion propositions and their holders. In *2004 AAAI Spring Symposium on Exploring Attitude and Affect in Text*, page 2224, 2004.

David Bindel, Jon Kleinberg, and Sigal Oren. How Bad is Forming Your Own Opinion? *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 57–66, October 2011.

Darius Braziunas. Pomdp solution methods. *University of Toronto*, 2003.

Eric Breck, Yejin Choi, and Claire Cardie. Identifying expressions of opinion in context. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 2683–2688. Morgan Kaufmann Publishers Inc., 2007.

Henry Brighton. Robust inference with simple cognitive models. *AAAI Spring Symposium: Cognitive Science Principles Meet AI-hard Problems*, (C. Lebiere & R. Wray (Eds.)):17–22, 2006.

Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. Deep blue. *Artificial Intelligence*, 134(1):57–83, 2002.

Charles S Carver. Self-awareness, perception of threat, and the expression of reactance through attitude change. *Journal of Personality*, 45(4):501–512, 1977.

Eugene Charniak. Toward a model of children's story comprehension. 1972.

Christy M.K. Cheung, Pui-Yee Chiu, and Matthew K.O. Lee. Online social networks: Why do students use facebook? *Computers in Human Behavior*, 27(4):1337–1343, July 2011.

Paul Chwelos, Izak Benbasat, and Albert Dexter, S. Research report: empirical test of an EDI adoption model. *Information Systems Research*, 12(3):304–321, 2001.

Mark J Clayton. Delphi: a technique to harness expert opinion for critical decision-making tasks in education. *Educational Psychology*, 17(4):373–386, 1997.

Michael D Cohen, James G March, and Johan P Olsen. A garbage can model of organizational choice. *Administrative science quarterly*, 17(1):1–25, 1972.

Bernard D Coleman and Victor J Mizel. On the general theory of fading memory. *Archive for Rational Mechanics and Analysis*, 29(1):18–31, 1968.

William D Crano. Primacy versus recency in retention of information and opinion change. *The Journal of Social Psychology*, (February 2013):37–41, 1977.

James P Curry, Douglas S Wakefield, James L Price, Charles W Mueller, and Joanne C McCloskey. Determinants of turnover among nursing department employees. *Research in Nursing & Health*, 8(4):397–411, 1985.

Jean Czerlinski, Gerd Gigerenzer, and Daniel Goldstein, G. How good are simple heuristics? In *Simple Heuristics that Make us Smart*. Oxford University Press, 1999.

Cristian Danescu-Niculescu-Mizil, Gueorgi Kossinets, Jon Kleinberg, and Lillian Lee. How opinions are received by online communities: a case study on amazon. com helpfulness votes. *Proceedings of the 18th International Conference on World Wide Web*, 2009.

Kahneman Daniel. *Thinking, fast and slow*. Farrar, Straus and Giroux, 2011.

Fred D Davis, Richard P Bagozzi, and Paul R Warshaw. User acceptance of computer technology: a comparison of two theoretical models. *Management Science*, 35(8): 982–1003, 1989.

Arthur P Dempster. Upper and lower probabilities induced by a multivalued mapping. *The Annuals of Mathmatical Statistics*, pages 325–339, 1967.

Franz Dietrich and Christian List. Opinion pooling on general agendas. (May):1–37, 2008.

Irene M Duhaime and Charles R Schwenk. Conjectures on cognitive simplification in acquisition and divestment decision making. *Academy of Management Review*, 10 (2):287–295, 1985.

James K Esser. Alive and well after 25 years: A review of groupthink research. *Organizational Behavior and Human Decision Processes*, 73(2/3):116–41, February 1998.

Gerd Gigerenzer and Henry Brighton. Homo Heuristicus: Why Biased Minds Make Better Inferences. *Topics in Cognitive Science*, 1(1):107–143, January 2009.

Gerd Gigerenzer and Wolfgang Gaissmaier. Heuristic decision making. *Annual review of psychology*, 62:451–82, January 2011. ISSN 1545-2085. doi: 10.1146/annurev-psych-120709-145346.

Gerd Gigerenzer and Daniel G Goldstein. Betting on one good reason: take the best and its relatives. In *Simple Heuristics that Make Us Smart*. Oxford University Press, 1999.

Thore Graepel, Ralf Herbrich, and Julian Gold. Learning to fight. In *Proceedings of the International Conference on Computer Games: Artificial Intelligence, Design and Education*, pages 193–200, 2004.

Simon Haykin and Neural Network. A comprehensive foundation. *Neural Networks*, 2, 2004.

Rainer Hegselmann and Ulrich Krause. Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of Artificial Societies and Social Simulation*, 5(3), 2002.

David R Heise. *Understanding events: Affect and the construction of social action.* Cambridge University Press, 1979.

Simon A Herbet. A behavioral model of rational choice. *The Quarterly Journal of Economics*, 69(1):99–118, 1955.

Jaakko Hintikka. Surface information and depth information. In *Information and Inference*, pages 263–297. Springer, 1970.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Valerie J Hoekstra. The Supreme Court and Opinion Change: An Experimental Study of the Court's Ability to Change Opinion. *American Politics Research*, 23 (1):109–129, January 1995.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1): 99–134, 1998.

Daniel Kahneman. Maps of Bounded Rationality:A Perspective on Intuitive Judgement and Choice. *Nobel Prize Lecture*, 8:251–401, 2002.

Daniel Kahneman and Amos Tversky. Subjective probability: A Judgment of Representativeness. *Cognitive psychology*, 3(3):430–454, 1972.

Daniel Kahneman and Amos Tversky. Choices, values, and frames. *The American Psychologist*, 39(4):341–350, 1984.

Daniel Kahneman, Paul Solvic, and Amos Tversky. *Judgment under uncertainty: Heuristics and biases.* Cambridge University Press, 1982.

Lauri Karttunen. Syntax and semantics of questions. *Linguistics and Philosophy*, 1 (1):3–44, 1977.

Herbert C Kelman. *Attitude change as a function of response restriction.* Tavistock, 1955.

Herbert C Kelman. Processes of Opinion Change. *Public Opinion Quarterly*, 25(1): 115–124, 1961.

Herbert C Kelman. Further thoughts on the processes of compliance, identification, and internalization. *Social Power and Political Influence*, pages 125–171, 1974.

George J. Klir and Bo Yuan. *Fuzzy sets and fuzzy logic*. Prentice Hall New Jersey, 1995.

Joachim Krueger. Personal beliefs and cultural stereotypes about racial characteristics. *Journal of Personality and Social Psychology*, 71(3):536, 1996.

Quoc V Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S Corrado, Jeff Dean, and Andrew Y Ng. Building high-level features using large scale unsupervised learning. In *Proceedings of International Conference in Machine Learning*, 2012.

Lawrence Leduc. Opinion change and voting behaviour in referendums. *European Journal of Political Research*, 41(6):711–732, October 2002.

Jill Fain Lehman, John Laird, Paul Rosenbloom, et al. A gentle introduction to soar, an architecture for human cognition. *Invitation to Cognitive Science*, 4:212–249, 1996.

Jill Fain Lehman, John Laird, and Paul Rosenbloom. A gentle introduction to soar, an architecture for human cognition: 2006 update. *University of Michigan*, 2006.

Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, volume 157, page 163, 1994.

Lorenzo Magnani. *Abduction, reason, and science: Processes of discovery and explanation*. Kluwer Academic/Plenum Publishers New York, 2001.

David Marr. *A computational investigation into the human representation and processing of visual information*. WH San Francisco: Freeman and Company, 1982.

Laura Martignon and Ulrich Hoffrage. Fast, frugal, and fit: Simple heuristics for paired comparison. *Theory and Decision*, pages 29–71, 2002.

Tara G Martin, Petra M Kuhnert, Kerrie Mengersen, and Hugh P Possingham. The power of expert opinion in ecological models using bayesian methods: impact of grazing on birds. *Ecological Applications*, 15(1):266–280, 2005.

Winter a Mason, Frederica R Conrey, and Eliot R Smith. Situating social influence processes: dynamic, multidirectional flows of influence within social networks. *Personality and social Psychology Review : an Official Journal of the Society for Personality and Social Psychology*, 11(3):279–300, August 2007.

William J McGuire. Personality and attitude change: An information-processing theory. *Psychological Foundations of Attitudes*, pages 171–196, 1968.

Ryszard S Michalski. A theory and methodology of inductive learning. *Artificial Intelligence*, 20(2):111–161, 1983.

Richard T Mowday, Richard M Steers, and Lyman W Porter. The measurement of organizational commitment. *Journal of Vocational Behavior*, 14(2):224–247, April 1979.

ST Mugford, Eamonn B Mallon, and Nigel R Franks. The accuracy of Buffon's needle: a rule of thumb used by ants to estimate area. *Behavioral Ecology*, 12(6):655–658, 2001.

Andrew Y Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 663–670, 2000.

Hå kan Nilsson, Peter Juslin, and Henrik Olsson. Exemplars in the mist: the cognitive substrate of the representativeness heuristic. *Scandinavian Journal of Psychology*, 49(3):201–12, June 2008.

Charles Egerton Osgood, William H May, and Murray S Miron. *Cross-cultural universals of affective meaning*. Urbana: University of Illinois Press, 1975.

Bo Pang and Lillian Lee. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008. ISSN 1554-0669.

Marion Petrie and Tim Halliday. Experimental and natural changes in the peacock's (Pavo cristatus) train can affect mating success. *Behavioral Ecology and Sociobiology*, 35(3):213–217, 1994.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*, volume 414. Wiley-Interscience, 2009.

Lance J Rips. *The psychology of proof: Deductive reasoning in human thinking*. The MIT Press, 1994.

Dawn T Robinson, Lynn Smith-Lovin, and Allison K Wisecup. Affect control theory. In *Handbook of the Sociology of Emotions*, pages 179–202. Springer, 2006.

Christabel L Rogalin, Shane D Soboroff, and Michael J Lovaglia. Power, status, and affect control. *Sociological Focus*, 40(2):202–220, 2007.

Tzachi Rosen, Solomon Eyal Shimony, and Eugene Santos Jr. Reasoning with bkbs – algorithms and complexity. *Annals of Mathematics and Artificial Intelligence*, 40 (3):403–425, 2004.

Stuart Jonathan Russell, Peter Norvig, Ernest Davis, Stuart Jonathan Russell, and Stuart Jonathan Russell. *Artificial Intelligence: a Modern Approach*, volume 2. Prentice hall Englewood Cliffs, 2010.

Eugene Santos, Hien Nguyen, Fei Yu, Keumjoo Kim, Deqing Li, John T Wilkinson, Adam Olson, and Russell Jacob. Intent-driven insider threat detection in intelligence analyses. In *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on*, volume 2, pages 345–349. IEEE, 2008.

Eugene Santos, Hien Nguyen, Fei Yu, Deqing Li, and John T Wilkinson. Impacts of analysts' cognitive styles on the analytic process. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pages 601–610. IEEE, 2010.

Eugene Santos, Hien Nguyen, Fei Yu, Keum Joo Kim, Deqing Li, John T Wilkinson, Adam Olson, Jacob Russell, and Brittany Clark. Intelligence analyses and the insider threat. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 42(2):331–347, 2012.

Eugene Santos Jr and Eugene S Santos. A framework for building knowledge-bases under uncertainty. *Journal of Experimental & Theoretical Artificial Intelligence*, 11(2):265–286, 1999.

Eugene Santos Jr, John T Wilkinson, and Eunice E Santos. Bayesian knowledge fusion. *Proceedings of the 22nd Florida Artificial Intelligence Research Society Conference*, (Laskey), 2009.

Ayse Pinar Saygin, Ilyas Cicekli, and Varol Akman. Turing test: 50 years later. In *The Turing Test*, pages 23–78. Springer, 2003.

Joseph L Schafer and John W Graham. Missing data: Our view of the state of the art. *Psychological Methods*, 7(2):147–177, 2002.

Greg Schmidt and Bernard Weiner. An attribution-affect-action theory of behavior replications of judgments of help-giving. *Personality and Social Psychology Bulletin*, 14(3):610–621, 1988.

Howard Schuman and Stanley Presser. The open and closed question. *American Sociological Review*, pages 692–712, 1979.

Charles R. Schwenk. Cognitive simplification processes in strategic decision-making. *Strategic Management Journal*, 5(December 1982):111–128, 2006.

Jagdish N Sheth. A model of industrial buyer behavior. *The Journal of Marketing*, pages 50–56, 1973.

Yoav Shoham, Rob Powers, and Trond Grenager. Multi-agent reinforcement learning: a critical survey. *Technical Report*, 2003.

Dean Keith Simonton. The Science of Genius. *Scientific American Mind*, pages 34–41, 2012.

Lynn Smith-Lovin. Impressions from events. *Journal of Mathematical Sociology*, 13 (1-2):35–70, 1987.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. Cambridge Univ Press, 1998.

Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

Alan M Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

Amos Tversky. Elimination by aspects: A theory of choice. *Psychological Review*, 79 (4):281, 1972.

Ralph B Vacchiano, Paul S Strauss, and Leonard Hochman. The open and closed mind: A review of dogmatism. *Psychological Bulletin*, 71(4):261, 1969.

Robert E Warren. Time and the spread of activation in memory. *Journal of Experimental Psychology: Human Learning and Memory*, 3(4):458, 1977.

Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards.* PhD thesis, University of Cambridge, 1989.

Duncan J Watts and Peter Sheridan Dodds. Influentials, networks, and public opinion formation. *Journal of Consumer Research*, 34(4):441–458, 2007.

William A Watts and Lewis E Holt. Persistence of opinion change induced under conditions of forewarning and distraction. *Journal of Personality and Social Psychology*, 37(5):778–789, 1979.

Gérard Weisbuch, Guillaume Deffuant, Frederic Amblard, and J-P Nadal. Interacting agents and continuous opinions dynamics. *Heterogenous Agents, Interactions and Economic Performance*, pages 225–242, 2003.

Joseph Weizenbaum. Elizaa computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1): 36–45, 1966.

Ping Xuan, Victor Lesser, and Shlomo Zilberstein. Communication decisions in multiagent cooperation: Model and experiments. In *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 616–623. ACM, 2001.

Ercan Yildiz, Daron Acemoglu, Asuman E. Ozdaglar, Amin Saberi, and Anna Scaglione. Discrete Opinion Dynamics with Stubborn Agents. *SSRN Electronic Journal*, pages 1–48, 2011.

Fei Yu and Eugene Santos Jr. Revisiting concepts of topicality and novelty-a new simple graph model that rewards and penalizes based on semantic links. In *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, pages 2656–2663. IEEE, 2012.

John R Zaller. *The nature and origins of mass opinion*. Cambridge university press, 1992.

Chapter 9

Appendices

9.1 Appendix A

Theorem 1. Let $K^X(\mathbf{L})$ (abbrev. K^X) be a base knowledge from source fragments $\{K_i^X : 1 \le i \le m, m \in \mathbb{N}\}$. We can derive the posterior probability for a feature as follows:

$$P_{K^X}(A = a) = \frac{v(A = a)}{m} + \frac{k}{|\mathbb{C}|m} \tag{9.1}$$

where $v(A)$ is the number of episodes in which feature A has a value assignment.

Proof. To compute the posterior probability of a feature A, we will need to discuss how the process of fusion changes its probability. In a source fragment, all features in the learning environment $\mathbf{L}$ except for the rooted feature are prior r.v.s. Its prior probability depends on the value assignment of A: If we have an explicit value, say $A = 1$, then the I-node $(A = 1)$ will have a prior probability 1.0. Otherwise if there is no value assignment for A, then we construct $|\mathbb{C}|$ I-nodes $(A = 1), (A = 2), \ldots, (A = |\mathbb{C}|)$. All these I-nodes will have the same prior probability $|\mathbf{C}|$.

Intuitively, we have $\sum_{j=1}^{|\mathbb{C}|} P(A = a_j) = 1$ for a source fragment. Now, let us discuss how their posterior probabilities change during fusion. During the process of fusion, each S-node will have a new I-node denoting the source. Thus, we just need to consider the sources that contain this r.v..

Case 1: There are $v(A = a)$ episodes with I-node $(A = a)$. The sum of joint probabilities is thus $\frac{v(A=a)}{m}$.

Case 2: There are k r.v.s with missing values. The sum of joint probabilities is thus $\frac{k}{|\mathbb{C}|}$.

Case 3: There are $m - k - v(A = a)$ episodes with no I-nodes $(A = a)$. The sum of joint probabilities is thus 0.

To sum up the posterior probabilities for the above three cases, we have:

$$P_{K^X}(A = a) = \frac{v(A = a)}{m} + \frac{k}{|\mathbb{C}|m} \tag{9.2}$$

$\square$

Lemma. We have the following property for a r.v.:

$$\sum_{i=1}^{|\mathbb{C}|} P_{K^X}(A = a_i) = 1 \tag{9.3}$$

Proof. Partition all the episodes in $\mathbf{L}$ into $\pi(\mathbf{L})$. Each cell in $\pi_{\mathbf{L}}(a)$ contains episodes with differerent values (including value ?) for feature A. We use $\pi_{\mathbf{L}}(a)$ denotes the cell containing value a for feature A. $|\pi_{\mathbf{L}}(a)|$ denotes the number of elements in the cell. We derive posterior probability for $P(A = a)$ according to Equation 9.1:

$$P_{K^X}(A = a) = \frac{|\pi_{\mathbf{L}}(A = a)|}{m} + \frac{k}{|\mathbb{C}|m}$$

To sum the posterior probability for all various values in $\mathbb{C}$, it becomes:

$$\sum_{i=1}^{|\mathbb{C}|} P_{K^X}(A = a_i) = \sum_{i=1}^{|\mathbb{C}|} \frac{|\pi_{\mathbf{L}}(A = a_i)|}{m} + \sum_{i=1}^{|\mathbb{C}|} \frac{k}{|\mathbb{C}|m}$$

$$= \sum_{i=1}^{|\mathbb{C}|} \frac{|\pi_{\mathbf{L}}(A = a_i)|}{m} + \frac{k|\mathbb{C}|}{|\mathbb{C}|m}$$

As we have $\sum_{i=1}^{|\mathbb{C}|} |\pi_{\mathbf{L}}(A = a_i)| = m - k$, the equation can be simplified to:

$$= \frac{m-k}{m} + \frac{k}{m}$$

$$= 1$$

□

Theorem 2. Let $K^X(\mathbf{L})$ (abbrev. K^X) be a base knowledge from source fragments $\{K_i^X : 1 \leq i \leq m, m \in \mathbb{N}\}$. We can derive joint probability of a feature along with a specification of source as the follows:

$$P_{K^X}(A = i, SA = j) = \frac{1}{m|\mathbb{C}|^{\mu}} \tag{9.4}$$

where $1 \leq j \leq m$, $1 \leq i \leq \mathbb{C}$.
μ is a parameter valued 0 if A in source fragment j has value assignment, valued 1 if A has no value assignments in source fragment j.

We can further derive the following probablistic answers:

Case 1: r.v. A has a value assignment i in source fragment j.

$$P_{K^X}(A = i, SA = j) = \frac{1}{m}$$

$$P_{K^X}(A = k, SA = j) = 0 \text{ where } k \neq i$$

Case 2: r.v. A has no value assignments in source fragment j.

$$\forall i,\ P_{K^X}(A = i, SA = j) = \frac{1}{m|\mathbb{C}|}$$

If we sum up all the joint probabilities over various instantiations of A, we derive the same answer for both cases:

$$P_{K^X}(SA = j) = \frac{1}{m}$$

Theorem 3. Let g be a subgraph of K^X that is compatible with l source fragments $\{K_i^X : 1 \leq i \leq l, l \in \mathbb{N}\}$. Among all r.v.s RV, we have k r.v.s with missing values and thus $n - k$ with value assignments. We have $0 \leq k \leq n$ and $k \in \mathbb{N}$. The result of belief updating inference is in the following form:

$$P_g(X) = \frac{l}{m} \prod_{a_i \in I_g, assn(a_i) \neq \{?\}} P(a_i) \tag{9.5}$$

The belief updating results over K^X consistent with X is invarient to the value of k.

Proof. Base case: Show that the statement holds for case $k = 0$.

k implies that all features have value assignments. In the process of constructing a single-rooted BKB from a learning episode $\mathbf{r}$, we only have one S-node connecting the value assignments of features (except root feature X) with X. In addition, the weight of this S-node is 1.0 as the root feature X has value assignment too (refer to Definition 5). When l fragments are fused together, additional I-nodes are inserted to distinguish the sources of original I-nodes and to avoid mutual exclusive problems. We compute belief updating as the following:

$$P_g(X) = \sum w_{r_i \in g, X \subset span(r_i)}(r_i)$$

$$= \sum_{i=1}^{l} w(SX = i) \prod_{a_i \in I_g, assn(a_i) \neq \{?\}} P(a_i)$$

According to Definition 6, all the weights for source I-nodes are computed as:

$$w(SX = i) = \frac{1}{m} \text{ for } 1 \leq i \leq l$$

Thus:

$$P_g(X) = \frac{l}{m} \prod_{a_i \in I_g, assn(a_i) \neq \{?\}} P(a_i)$$

Inductive Step: Assume the statement holds for case $k = t$.

For $k = t + 1$, g now has one additional r.v. with missing values, say A, compared to case where $k = t$. Let $\pi(A)$ denote the set of all instantiations of A, we should have $|\pi(A)| = \mathbb{C}$. We compute belief updating as the following:

$$P_g(X)' = \sum w'_{r_i \in g, X \subset span(r_i)}(r_i)$$

$$= \sum w_{r_i \in g, X \subset span(r_i)}(r_i) \sum_{j=1}^{\mathbb{C}} P(A = a_j)$$

We have:

$$\sum_{j=1}^{\mathbb{C}} P(A = a_j) = 1$$

Thus:

$$P_g(X)' = \sum w_{r_i \in g, X \subset span(r_i)}(r_i)$$
$$= \frac{l}{m} \prod_{a_i \in I_g, assn(a_i) \neq \{?\}} P(a_i)$$

□

Lemma. Belief updating inference sums up to 1 for the following special case: all r.v.s in each learning episode have missing values.

$$P_{K^X}(X) = 1$$

Proof. According to Definition , K^X has one subgraph g compatible with all the learning episodes. Thus, belief updating result becomes the following:

$$P_g(X) = \frac{m}{m} = 1$$

□

Lemma. If g has modeled multiple states of root r.v. X, the posterior probability for each state is the same.

$$P_g(X = x_i) = \frac{l}{m|\mathbb{C}|} \prod_{a_i \in I_g, assn(a_i) \neq \{?\}} P(a_i) \tag{9.6}$$

where $1 \leq i \leq \mathbb{C}$.

Proof. According to Definition 5, all instantiations of the root r.v. has the same probablity. Thus, the belief updating result for each instantiation the same. □

Theorem 4 (Joint Probability)**.** Let g be a subgraph of K^X that is compatible with l source fragments $\{K_i^X : 1 \leq i \leq l, l \in \mathbb{N}\}$. Among all r.v.s RV, we have k r.v.s with missing values and thus $n - k$ with value assignments. We have $0 \leq k \leq n$ and $k \in \mathbb{N}$.

The posterior probabilities for multiple target r.v.s can be derived from the closed-form solution of $P_g(X)$. First, we partition U over all features into π'. Each cell $\sigma' \in \pi'$ contain evidence provided on a feature and evidence provided on the source of a feature. An example of a cell can be $\{A = i, SA = j\}$ where i is the value assignment on r.v. A and j is the name of the source fragment containing

$A = i$. We go over each cell σ', and computing joint probability with the evidences provided in that cell. To make it clear, our objective is to derive the joint probability in the form $P_g(X = x_i, U)$. We derive the joint probability recursively by deriving an incremental solution $P'_g(X = x_i, U)$ by considering every $\sigma' \in \pi'$. The full Algorithm 9.3 can be found in Appendix B.

9.2 Appendix B

Input: Learning episodes mtx of size $m \times n$
Input: Features $feats$ of size $n \times 1$
Input: Tasks $tasks$ consists z states each of size $y \times 2$
Input: Number of classes c
Input: Index of root feature r
Output: Joint probabilities $results$ of size $z \times 1$

```
var rvs ← initRVs(feats);
var pa ← initPA(mtx, c);
for i = 0 to z do
    var sum ← 0;
    for j ← 0 to m do
        var part ← 1/m;
        for k ← 0 to n do
            var t ← computeJointProbability(tasks(i), j, k, mtx, pa, r, rvs);
            if t = 0 then
                part ← 0;
                break;
            else
                part ← part × t;
        sum = sum + p;
    results[i]=sum;
```

Figure 9.1: Algorithm to Compute Joint Probabilities in Base Knowledge K^X

```
Input: Learning episodes mtx of size m × n
Input: States sts of size y × 2
Input: RVs rvs of size n × 2
Input: Marginal probabilities for features pa of size n × c
Input: Index of the feature k
Input: Index of the episode j
Input: Index of the root feature r
Output: Joint probability p

var sptr ← −1;
var fptr ← −1;
var term ← 1.0;
t ← findAssignments(sts, k);
if length(t) = 0 then
    if mtx[j, k] = −1 then
        return 1;
    else
        return pa[k, mtx[j, k]];

for i ← 0 to length(t) do
    if isRootFeature(t[i][0])=true or supportsRootFeature(t[i][0)=true then
        continue;
    if isSourceFeature(t[i][0])=true then
        sptr ← i;
    fptr ← i;

if doMatch(t)=false then
    return 0;

if length(t)=1 and isFeature(t[0, 0]) then
    if isRootFeature(t[0, 0]) then
        if mtx[m, t[0, 0]] = −1 then
            return 1/c;
        else
            return 1;
    return pa[t[0, 0], t[0][1]];
```

Figure 9.2: Algorithm to Compute Joint Probability (Part I)

```
if mtx[t[sptr, 1], getSupportedFeature(t[sptr, 0])] = −1 then
    if mtx[j, getSupportedFeature(t[sptr, 0])] = −1 then
        term ← term × 1/m;
    else
        term ← term × 1/(m×c);
    if fptr = −1 then
        return term;
    if mtx[j, t[fptr, 0]] = −1 then
        return term × 1/c;
    else if mtx[j, t[fptr, 0]] = mtx[t[sptr, 1], getSupportedFeature(t[sptr, 0])]
    then
        return term;
    else
        return 0;
if mtx[t[sptr, 1], getSupportedFeature(t[sptr, 0])] =
mtx[j, getSupportedFeature(t[sptr, 0])] then
    term ← term × 1/m;
else
    return 0;
if fptr = −1 then
    return term;
if mtx[j, t[fptr, 0]] = mtx[t[sptr, 1], getSupportedFeature(t[sptr, 0])] then
    return term;
else
    return 0;
```

Figure 9.3: Algorithm to Compute Joint Probability (Part II)

9.3 Appendix C

9.3.1 Idealistic-Active vs. Malleable-Active Agent

Court is a classic environment in which many IA-MA style decision making processes take place to influence judical opinions. Lawyers are IA-style agents seeking the best methods to convince the judges and judges are MP-style agents who are required by laws to minimize their influence on opinion change. According to Hoekstra (Hoekstra, 1995), the court does not need to run for re-election so it does not have self-interested motives regarding the direction of its rulings. An idealistic-active agent seeks to persuade the other agent and has no interest in changing its own opinion. At each time step, the primary IA-style agent makes a decision on the knowledge to be recommeneded to the other agent. For example, an IA agent can suggest that the other agent discard evidence that the IA agent considers irrelvant. Simiarly, an IA-style agent can suggest that the other agent include information that was previously neglected, or do nothing if there is nothing an IA agent can do to decrease the gap between them.

When the communication is single directional (e.g. teaching), the decision-making problem can be formulated as between two IA-MA agents. One may analyze the source of information that results in differences in their opinions, and then communicate it to the MA agent so that the MA agent can obtain the same understanding of the world as the IA agent.

A state of the environment $s \in S$ is a 3-tuple (K_1^t, K_2^t, k_{12}^t) where K_1^t is the knowledge base relied upon by agent e_1 at time t, and K_2^t is the knowledge base relied upon by agent e_2 at time t. k_{12}^t represents the knowledge communicated from e_1 to e_2 at time t and can be either empty or contain one learning episode. Set S captures all possible combinations of knowledge agent e_1 and e_2 may use to derive their opinions, and information e_1 can communicate at a given time formally defined as

$$S = V_1^K \times V_2^K \times \cup\{v_i\}$$

where $|v_i| = 1$, $v_i \in V_1^K$.

In the case of an IA-MA interaction, the malleable-passive agent needs to consider actions from the other IA agent: pasadd, pasremove in addition to the actions it initiates on its own (intadd, intremove, donothing). On the other hand, an IA agent considers actions: intadd, intremove, donothing, actadd, and actremove. For both of the agents, actions such as intadd, intremove are originated from the agent itself (thus referred to as internal influence). Actions such as actadd and actremove are actions performed on the other agent (thus become external influence to the other agent). Lastly, actions such as pasadd and pasremove are actions received from the other agent (thus also external influences). The exact learning episode for an action depends on the learning episodes included in the knowledge base currently in use. As the action of one agent depends on the other agent's style, each action function depends on both agents' knowledge base in use.

$$A_1(s^t) = A_1(K_1^t, K_2^t)$$

and

$$A_2(s^t) = A_2(K_2^t, K_2^t)$$

We define the **action function** for e_1 as

$$a \in A_1(s^t) \text{ if: } \begin{cases} \mathit{type}(a) \text{ is intadd, and } \mathit{value}(a) \notin K_1^t \\ \mathit{type}(a) \text{ is intremove, and } \mathit{value}(a) \in K_1^t \\ \mathit{type}(a) \text{ is donothing, and } \mathit{value}(a) = \phi \\ \mathit{type}(a) \text{ is actadd, and } \mathit{value}(a) \in K_1^t \\ \mathit{type}(a) \text{ is actremove, and } \mathit{value}(a) =\in K_1^t \end{cases}$$

We define the **action function** for e_2 as

$$a \in A_2(s^t) \text{ if: } \begin{cases} \mathit{type}(a) \text{ is intadd, and } \mathit{value}(a) \notin K_2^t \\ \mathit{type}(a) \text{ is intremove, and } \mathit{value}(a) \in K_2^t \\ \mathit{type}(a) \text{ is donothing, and } \mathit{value}(a) = \phi \\ \mathit{type}(a) \text{ is pasadd, and } \mathit{value}(a) \notin K_2^t \\ \mathit{type}(a) \text{ is pasremove, and } \mathit{value}(a) =\in K_2^t \end{cases}$$

The computation of the **transition probability** is broken down into three parts: the first part checks the validity of transition for agent e_1, the second part checks the validity of transition for agent e_2, the third part checks whether two actions are compatible and valid.

$$P(s_t, a_1, s_{t+1}) = \begin{cases} 1.0 \text{ if } P(K_1^t, a_1, K_1^{t+1}) = 1,\ P(K_2^t, a_2^*, K_2^{t+1}) = 1, \text{ two actions are compatible} \\ 0.0 \text{ otherwise} \end{cases}$$

where $P(K_1^t, a_1, K_1^{t+1})$ is the probability of resulting in a knowledge base K_1^{t+1} by applying change a_1 to current knowledge base K_1^t. $P^(K_2^t, a_2^*, K_2^{t+1})$ is the probability of resulting in a knowledge base K_2^{t+1} by applying change a_2^* to current knowledge base K_2^t. For instance, if agent e_1 does nothing ($\mathit{type}(a_1) = \mathit{donothing}$), then we should have $K_1^t = K_1^{t+1}$. We assume the environment to be non-stochastic (e.g. information-processing produce is always correct); that is, a valid knowledge transformation action has a probability of 1.0. We formally specify the transition probability as follows:

$$P(K_1^t, a_1, K_1^{t+1}) = \begin{cases} 1.0 \text{ if one of the conditions below holds} \\ 0.0 \text{ otherwise} \end{cases}$$

- $\mathit{type}(a_1) = \mathit{donothing}, K_1^{t+1} = K_1^t$

- $type(a_1) = intadd, K_1^{t+1} = K_1^t \cup value(a_1)$
- $type(a_1) = intremove, K_1^t = K_1^{t+1} \cup value(a_1)$
- $type(a_1) = actadd, K_1^t = K_1^{t+1}$
- $type(a_1) = actremove, K_1^t = K_{t+1}$

and

$$P(K_2^t, a_2^*, K_2^{t+1}) = \begin{cases} 1.0 \text{ if one of the conditions below holds} \\ 0.0 \text{ otherwise} \end{cases}$$

- $a_2^* = \max\limits_{a \in A_2(s^t)} (R_a(s, s')), type(a_2^*) = donothing, K_2^{t+1} = K_2^t$
- $a_2^* = \max\limits_{a \in A_2(s^t)} (R_a(s, s')), type(a_2^*) = intadd, K_2^{t+1} = K_2^t \cup value(a_2^*)$
- $a_2^* = \max\limits_{a \in A_2(s^t)} (R_a(s, s')), type(a_2^*) = intremove, K_2^t = K_2^{t+1} \cup value(a_2^*)$
- $a_2^* = \max\limits_{a \in A_2(s^t)} (R_a(s, s')), type(a_2^*) = pasadd, K_2^{t+1} = K_2^t \cup value(a_2^*)$
- $a_2^* = \max\limits_{a \in A_2(s^t)} (R_a(s, s')), type(a_2^*) = pasremove, K_2^t = K_2^{t+1} \cup value(a_2^*)$

where a_2^* is the greedy action e_2 decide on according to its reward function. Lastly, two actions a_1 and a_2^* need to be compatible with each other.

(a_1, a_2^*) are compatible if

- $type(a_1)$=donothing or
- $type(a_1)$=intadd or
- $type(a_1)$=intremove or
- $type(a_1)$=actadd, $type(a_2^*)$=pasadd
  $value(k_{12}) = value(a_2^*)$, and $value(a_1) = value(a_2^*)$ or
- $type(a_1)$=actremove, $type(a_2^*)$=pasremove
  $value(k_{12}) = value(a_2^*)$, and $value(a_1) = value(a_2^*)$

The **reward function** for e_1 is derived from the goal function as

$$R_a(s, s') = \begin{cases} -|o_1^{t+1} - o_2^{t+1}| \text{ if } o_1^{t+1} = o_1^t \\ -1 \text{ otherwise} \end{cases}$$

The **reward function** for e_2 is derived from the goal function as

$$R_a(s, s') = -|o_1^{t+1} - o_2^{t+1}|$$

Now, we have provided a complete definition of the decision process (states, action function, reward function, and probability transition function) for an IA-MA case.

9.3.2 Malleable-Active vs. Idealistic-Active Agent

When the communication is single directional (e.g. teaching), the decision-making problem can be formulated as between two MA-IA agents. In this case, a MA agent compares among information it possesses along with the information recommended by the other agent to address the gap in their opinions. The MA-IA and IA-MA cases are different in terms of whose decision-making process we model. In the MA-IA case, we model the MA agent's decision-making process. In the IA-MA case, we model the IA agent's decision-making process.

A state of the environment $s \in S$ is a 3-tuple (K_1^t, K_2^t, k_{21}^t) where K_1^t is the knowledge base relied upon by agent e_1 at time t, and K_2^t is the knowledge base relied upon by agent e_2 at time t. k_{21}^t represents the knowledge communicated from e_2 to e_1 at time t and can be either empty or contain one learning episode. Set S captures all possible combinations of knowledge agent e_1 and e_2 may use to derive their opinions, and information e_2 can communicate at a given time formally defined as

$$S = V_1^K \times V_2^K \times \cup\{v_i\}$$

where $|v_i| = 1$, $v_i \in V_2^K$ and $v_i = value(a_2^*)$. The information sent from agent e_2 to agent e_1 needs to contain the same information as in e_2's greedy action otherwise the state of environment is invalid as such state would never occur.

In the case of a MA-IA interaction, the malleable-active agent needs to consider actions from the IA agent: pasadd, pasremove in addition to the actions it initiates on its own (intadd, intremove, and donothing). On the other hand, an IA agent considers actions: intadd, intremove, donothing, actadd, and actremove. The exact learning episode for an action depends on the learning episodes included in the knowledge base currently in use. Again as the action of one agent depends on the other agent's style, each action function depends on both agents' knowledge base in use.

$$A_1(s^t) = A_1(K_1^t, K_2^t)$$

and

$$A_2(s^t) = A_2(K_2^t, K_2^t)$$

We define the **action function** for e_1 as

$$a \in A_1(s^t) \text{ if: } \begin{cases} type(a) \text{ is intadd, and } value(a) \notin K_1^t \\ type(a) \text{ is intremove, and } value(a) \in K_1^t \\ type(a) \text{ is donothing, and } value(a) = \phi \\ type(a) \text{ is pasadd, and } value(a) \notin K_1^t \\ type(a) \text{ is pasremove, and } value(a) =\in K_1^t \end{cases}$$

We define the **action function** for e_2 as

$$a \in A_2(s^t) \text{ if: } \begin{cases} type(a) \text{ is intadd, and } value(a) \notin K_2^t \\ type(a) \text{ is intremove, and } value(a) \in K_2^t \\ type(a) \text{ is donothing, and } value(a) = \phi \\ type(a) \text{ is actadd, and } value(a) \in K_2^t \\ type(a) \text{ is actremove, and } value(a) = \in K_2^t \end{cases}$$

The computation of the **transition probability** is broken down into three parts: the first part checks the validity of transition for agent e_1, the second part checks the validity of transition for agent e_2, the third part checks whether two actions are compatible and valid.

$$P(s_t, a_1, s_{t+1}) = \begin{cases} 1.0 \text{ if } P(K_1^t, a_1, K_1^{t+1}) = 1,\ P(K_2^t, a_2^*, K_2^{t+1}) = 1, \text{ two actions are compatible} \\ 0.0 \text{ otherwise} \end{cases}$$

where $P(K_1^t, a_1, K_1^{t+1})$ is the probability of resulting in a knowledge base K_1^{t+1} by applying change a_1 to current knowledge base K_1^t. $P^(K_2^t, a_2^*, K_2^{t+1})$ is the probability of resulting in a knowledge base K_2^{t+1} by applying change a_2^* to current knowledge base K_2^t. For instance, if agent e_1 does nothing ($type(a_1) = donothing$), then we should have $K_1^t = K_1^{t+1}$. We assume the environment to be non-stochastic (e.g. information-processing produce is always correct); that is, a valid knowledge transformation action has a probability of 1.0. We formally specify the transition probability as:

$$P(K_1^t, a_1, K_1^{t+1}) = \begin{cases} 1.0 \text{ if one of the conditions below holds} \\ 0.0 \text{ otherwise} \end{cases}$$

- $type(a_1) = donothing, K_1^{t+1} = K_1^t$
- $type(a_1) = intadd, K_1^{t+1} = K_1^t \cup value(a_1)$
- $type(a_1) = intremove, K_1^t = K_1^{t+1} \cup value(a_1)$
- $type(a_1) = pasadd, K_1^t \cup value(a_1) = K_1^{t+1}$
- $type(a_1) = pasremove, K_1^t = K_{t+1} \cup value(a_1)$

and

$$P(K_2^t, a_2^*, K_2^{t+1}) = \begin{cases} 1.0 \text{ if one of the conditions below holds} \\ 0.0 \text{ otherwise} \end{cases}$$

- $a_2^* = \max_{a \in A_2(s^t)} (R_a(s, s')), type(a_2^*) = donothing, K_2^{t+1} = K_2^t$
- $a_2^* = \max_{a \in A_2(s^t)} (R_a(s, s')), type(a_2^*) = intadd, K_2^{t+1} = K_2^t \cup value(a_2^*)$
- $a_2^* = \max_{a \in A_2(s^t)} (R_a(s, s')), type(a_2^*) = intremove, K_2^t = K_2^{t+1} \cup value(a_2^*)$

- $a_2^* = \max_{a \in A_2(s^t)} (R_a(s, s')), type(a_2^*) = actadd, K_2^{t+1} = K_2^t$

- $a_2^* = \max_{a \in A_2(s^t)} (R_a(s, s')), type(a_2^*) = actremove, K_2^t = K_2^{t+1}$

where a_2^* is the greedy action e_2 decide on according to its reward function. Lastly, two actions a_1 and a_2^* need to be compatible with each other.

(a_1, a_2^*) are compatible if

- $type(a_1)$=donothing or
- $type(a_1)$=intadd or
- $type(a_1)$=intremove or
- $type(a_1)$=pasadd, $type(a_2^*)$=actadd
  $value(k_{21}) = value(a_2^*)$, and $value(a_1) = value(a_2^*)$ or
- $type(a_1)$=pasremove, $type(a_2^*)$=actremove
  $value(k_{21}) = value(a_2^*)$, and $value(a_1) = value(a_2^*)$

The **reward function** for e_1 is derived from the goal function as

$$R_a(s, s') = -|o_1^{t+1} - o_2^{t+1}|$$

The **reward function** for e_2 is derived from the goal function as

$$R_a(s, s') = \begin{cases} -|o_1^{t+1} - o_2^{t+1}| \text{ if } o_2^{t+1} = o_2^t \\ -1 \text{ otherwise} \end{cases}$$

9.3.3 Idealistic-Active vs. Idealistic-Active Agent

When the communication is bi-directional (e.g. shared decision-making between a physician-patient pair), the decision-making problem can be formulated as between two IA-IA agents. However as an IA agent has no interests in changing its own opinion, a conversation between two IA agents is the worst case for reaching opinion consensus.

A state of the environment $s \in S$ is a 4-tuple $(K_1^t, K_2^t, k_{21}^t, k_{12}^t)$ where K_1^t is the knowledge base relied upon by agent e_1 at time t, and K_2^t is the knowledge base relied upon by agent e_2 at time t. k_{12}^t represents the knowledge communicated from e_1 to e_2 at time t and can be either empty or contain one learning episode. k_{21}^t represents the knowledge communicated from e_2 to e_1 at time t and can be either empty or contain one learning episode. Set S captures all possible combinations of knowledge agent e_1 and e_2 may use to derive their opinions, and information e_2 and e_1 can communicate at a given time. It is formally defined as

$$S = V_1^K \times V_2^K \times \cup v_i \times \cup v_j$$

where $|v_i| = 1$, $|v_j| = 1$, $v_j \in V_1^K$, and $v_i \in V_2^K$.

In the case of an IA-IA interaction, an IA agent considers actions: intadd, intremove, donothing, actadd, and actremove. However, since an IA agent has complete knowledge about the other agent, it is aware that the other agent would not accept any information from it. The exact learning episode for an action depends on the learning episodes included in the knowledge base currently in use. Similar with the MP-MP case, the action function for both agents only depends on its own domain knowledge at the time

$$A_1(s^t) = A_1(K_1^t)$$

and

$$A_2(s^t) = A_2(K_2^t)$$

We define the **action function** for e_1 and e_2 as

$$a \in A_i(s^t) \text{ that } i \in [1, 2] \text{ if: } \begin{cases} type(a) \text{ is intadd, and } value(a) \notin K_i^t \\ type(a) \text{ is intremove, and } value(a) \in K_i^t \\ type(a) \text{ is donothing, and } value(a) = \phi \\ type(a) \text{ is actadd, and } value(a) \in K_i^t \\ type(a) \text{ is actremove, and } value(a) = \in K_i^t \end{cases}$$

The computation of the **transition probability** is broken down into three parts: the first part checks the validity of transition for agent e_1, the second part checks the validity of transition for agent e_2, the third part checks whether two actions are compatible and valid.

$$P(s_t, a_1, s_{t+1}) = \begin{cases} 1.0 \text{ if } P(K_1^t, a_1, K_1^{t+1}) = 1,\ P(K_2^t, a_2^*, K_2^{t+1}) = 1, \text{ two actions are compatible} \\ 0.0 \text{ otherwise} \end{cases}$$

where $P(K_1^t, a_1, K_1^{t+1})$ is the probability of resulting in a knowledge base K_1^{t+1} by applying change a_1 to current knowledge base K_1^t. $P^{(}K_2^t, a_2^*, K_2^{t+1})$ is the probability of resulting in a knowledge base K_2^{t+1} by applying change a_2^* to current knowledge base K_2^t. For instance, if agent e_1 does nothing ($type(a_1) = donothing$), then we should have $K_1^t = K_1^{t+1}$. We assume the environment to be non-stochastic (e.g. information-processing produce is always correct); that is, a valid knowledge transformation action has a probability of 1.0. We formally specify the transition probability as follows:

$$P(K_1^t, a_1, K_1^{t+1}) = \begin{cases} 1.0 \text{ if one of the conditions below holds} \\ 0.0 \text{ otherwise} \end{cases}$$

- $type(a_1) = donothing, K_1^{t+1} = K_1^t$
- $type(a_1) = intadd, K_1^{t+1} = K_1^t \cup value(a_1)$

- $type(a_1) = intremove, K_1^t = K_1^{t+1} \cup value(a_1)$

and

$$P(K_2^t, a_2^*, K_2^{t+1}) = \begin{cases} 1.0 \text{ if one of the conditions below holds} \\ 0.0 \text{ otherwise} \end{cases}$$

- $a_2^* = \max\limits_{a \in A_2(s^t)} (R_a(s, s')), type(a_2^*) = donothing, K_2^{t+1} = K_2^t$
- $a_2^* = \max\limits_{a \in A_2(s^t)} (R_a(s, s')), type(a_2^*) = intadd, K_2^{t+1} = K_2^t \cup value(a_2^*)$
- $a_2^* = \max\limits_{a \in A_2(s^t)} (R_a(s, s')), type(a_2^*) = intremove, K_2^t = K_2^{t+1} \cup value(a_2^*)$

where a_2^* is the greedy action e_2 decide on according to its reward function. Lastly, two actions a_1 and a_2^* need to be compatible with each other.

(a_1, a_2^*) are compatible if

- $type(a_1)$=donothing or
- $type(a_1)$=intadd or
- $type(a_1)$=intremove or

The **reward function** for e_1 is derived from the goal function as

$$R_a(s, s') = \begin{cases} -|o_1^{t+1} - o_2^{t+1}| \text{ if } o_1^{t+1} = o_1^t \\ -1 \text{ otherwise} \end{cases}$$

The **reward function** for e_2 is derived from the goal function as

$$R_a(s, s') = \begin{cases} -|o_1^{t+1} - o_2^{t+1}| \text{ if } o_2^{t+1} = o_2^t \\ -1 \text{ otherwise} \end{cases}$$

9.3.4 Mixed-Goal vs. Mixed-Goal Agent

Now we consider a situation where each agent is both adaptive and persuasive. Thus, we have malleability-idealism scale $\gamma \in (0, 1)$ and passivity-activism scale $\zeta \in (0, 1)$.

A state of the environment $s \in S$ is a 4-tuple $(K_1^t, K_2^t, k_{21}^t, k_{12}^t)$ where K_1^t is the knowledge base relied upon by agent e_1 at time t, and K_2^t is the knowledge base relied upon by agent e_2 at time t. k_{12}^t represents the knowledge communicated from e_1 to e_2 at time t and can be either empty or contain one learning episode. k_{21}^t represents the knowledge communicated from e_2 to e_1 at time t and can be either empty or contain one learning episode. Set S captures all possible combinations of knowledge agent e_1 and e_2 may use to derive their opinions, and information e_2 and

e_1 can communicate at a given time. It is formally defined as

$$S = V_1^K \times V_2^K \times \cup v_i \times \cup v_j$$

where $|v_i| = 1$, $|v_j| = 1$, $v_j \in V_1^K$, and $v_i \in V_2^K$.

In the case of an interaction between two agents with mixed-goals, each agent considers seven actions: intadd, intremove, donothing, actadd, actremove, pasadd, and pasremove. The exact learning episode for an action depends on the learning episodes included in the knowledge base currently in use. The action function for both agents only depends on both its domain knowledge but also the other agent's domain knowledge

$$A_1(s^t) = A_1(K_1^t, K_2^t)$$

and

$$A_2(s^t) = A_2(K_2^t, K_2^t)$$

We define the **action function** for e_1 as

$$a \in A_1(s^t) \text{ if: } \begin{cases} \mathit{type}(a) \text{ is intadd, and } \mathit{value}(a) \notin K_1^t \\ \mathit{type}(a) \text{ is intremove, and } \mathit{value}(a) \in K_1^t \\ \mathit{type}(a) \text{ is donothing, and } \mathit{value}(a) = \phi \\ \mathit{type}(a) \text{ is actadd, and } \mathit{value}(a) \in K_1^t \\ \mathit{type}(a) \text{ is actremove, and } \mathit{value}(a) = \in K_1^t \\ \mathit{type}(a) \text{ is pasadd, and } \mathit{value}(a) \in K_2^t \\ \mathit{type}(a) \text{ is pasremove, and } \mathit{value}(a) = \in K_2^t \end{cases}$$

We define the **action function** for e_2 as

$$a \in A_2(s^t) \text{ if: } \begin{cases} \mathit{type}(a) \text{ is intadd, and } \mathit{value}(a) \notin K_2^t \\ \mathit{type}(a) \text{ is intremove, and } \mathit{value}(a) \in K_2^t \\ \mathit{type}(a) \text{ is donothing, and } \mathit{value}(a) = \phi \\ \mathit{type}(a) \text{ is actadd, and } \mathit{value}(a) \in K_2^t \\ \mathit{type}(a) \text{ is actremove, and } \mathit{value}(a) = \in K_2^t \\ \mathit{type}(a) \text{ is pasadd, and } \mathit{value}(a) \in K_1^t \\ \mathit{type}(a) \text{ is pasremove, and } \mathit{value}(a) = \in K_1^t \end{cases}$$

The computation of the **transition probability** is broken down into three parts: the first part checks the validity of transition for agent e_1, the second part checks the validity of transition for agent e_2, the third part checks whether two actions are compatible and valid

$$P(s_t, a_1, s_{t+1}) = \begin{cases} 1.0 \text{ if } P(K_1^t, a_1, K_1^{t+1}) = 1,\ P(K_2^t, a_2^*, K_2^{t+1}) = 1, \text{ two actions are compatible} \\ 0.0 \text{ otherwise} \end{cases}$$

where $P(K_1^t, a_1, K_1^{t+1})$ is the probability of resulting in a knowledge base K_1^{t+1} by applying change a_1 to current knowledge base K_1^t. $P^(K_2^t, a_2^*, K_2^{t+1})$ is the probability of resulting in a knowledge base K_2^{t+1} by applying change a_2^* to current knowledge base K_2^t. For instance, if agent e_1 does nothing ($type(a_1) = donothing$), then we should have $K_1^t = K_1^{t+1}$. We assume the environment to be non-stochastic (e.g. information-processing produce is always correct); that is, a valid knowledge transformation action has a probability of 1.0. We formally specify the transition probability as the following:

$$P(K_1^t, a_1, K_1^{t+1}) = \begin{cases} 1.0 \text{ if one of the conditions below holds} \\ 0.0 \text{ otherwise} \end{cases}$$

- $type(a_1) = donothing, K_1^{t+1} = K_1^t$
- $type(a_1) = intadd, K_1^{t+1} = K_1^t \cup value(a_1)$
- $type(a_1) = intremove, K_1^t = K_1^{t+1} \cup value(a_1)$
- $type(a_1) = actadd, K_1^{t+1} = K_1^t$
- $type(a_1) = actremove, K_1^{t+1} = K_1^t$
- $type(a_1) = pasadd, K_1^{t+1} = K_1^t \cup value(a_1)$
- $type(a_1) = pasremove, K_1^t = K_1^{t+1} \cup value(a_1)$

and

$$P(K_2^t, a_2^*, K_2^{t+1}) = \begin{cases} 1.0 \text{ if one of the conditions below holds} \\ 0.0 \text{ otherwise} \end{cases}$$

- $a_2^* = \max\limits_{a \in A_2(s^t)}(R_a(s, s')), type(a_2^*) = donothing, K_2^{t+1} = K_2^t$
- $a_2^* = \max\limits_{a \in A_2(s^t)}(R_a(s, s')), type(a_2^*) = intadd, K_2^{t+1} = K_2^t \cup value(a_2^*)$
- $a_2^* = \max\limits_{a \in A_2(s^t)}(R_a(s, s')), type(a_2^*) = intremove, K_2^t = K_2^{t+1} \cup value(a_2^*)$
- $a_2^* = \max\limits_{a \in A_2(s^t)}(R_a(s, s')), type(a_2^*) = actadd, K_2^{t+1} = K_2^t$
- $a_2^* = \max\limits_{a \in A_2(s^t)}(R_a(s, s')), type(a_2^*) = actremove, K_2^{t+1} = K_2^t$
- $a_2^* = \max\limits_{a \in A_2(s^t)}(R_a(s, s')), type(a_2^*) = pasadd, K_2^{t+1} = K_2^t \cup value(a_2^*)$
- $a_2^* = \max\limits_{a \in A_2(s^t)}(R_a(s, s')), type(a_2^*) = pasremove, K_2^t = K_2^{t+1} \cup value(a_2^*)$

where a_2^* is the greedy action e_2 decide on according to its reward function. Lastly, two actions a_1 and a_2^* need to be compatible with each other.

(a_1, a_2^*) are compatible if

- $type(a_1)$=donothing or
- $type(a_1)$=intadd or
- $type(a_1)$=intremove or
- $type(a_1)$=actadd, $type(a_2^*)$=pasadd
  $value(k_{12}) = value(a_2^*)$, $value(k_{21}) = \phi$, and $value(a_1) = value(a_2^*)$ or
- $type(a_1)$=actremove, $type(a_2^*)$=pasremove
  $value(k_{12}) = value(a_2^*)$, $value(k_{12}) = \phi$, and $value(a_1) = value(a_2^*)$ or
- $type(a_1)$=pasadd, $type(a_2^*)$=actadd
  $value(k_{21}) = value(a_2^*)$, $value(k_{12}) = \phi$, and $value(a_1) = value(a_2^*)$ or
- $type(a_1)$=pasremove, $type(a_2^*)$=actremove
  $value(k_{21}) = value(a_2^*)$, $value(k_{12} = \phi$, and $value(a_1) = value(a_2^*)$

The **reward function** for both e_1 and e_2 is derived from the goal function as

$$R_a(s, s') = -\zeta|o_1^{t+1} - o_2^{t+1}| - \gamma|o_1^t - o_1^{t+1}|$$

where $\gamma \in (0, 1)$ is the malleability-idealism scale and $\zeta \in (0, 1)$ is the passivity-activism scale.