

## 6 Conclusion

### 6.1 Summary of Findings

This thesis presents a novel system identification/modeling method—*Turing Learning* for inferring agent behavior. Through competitive coevolution between classifiers and models, the method does not rely on predefined metrics for measuring the difference between the agents and models. Instead, the classifiers substitute the metrics, which are incorporated into the evolutionary process. During the evolutionary process, the classifiers only need to output a Boolean value indicating whether the observed individual is an agent or model.

The merits of *Turing Learning* were demonstrated through successfully inferring various agent behaviors ranging from swarm behaviors to deterministic/stochastic behaviors of a single agent. When inferring an unknown swarm behavior, it would be challenging to quantitatively measure the difference (e.g., motion) between the models and agents using predefined metrics, due to the interaction within agents and between agents and environment. The motion of each agent in the swarm appears to be stochastic. We have shown that through evolving classifiers that only need to distinguish between the observed agents and models, it is sufficient to infer the behavioral rules of the swarming agents. In other words, in order to ‘trick’ the classifiers to judge them as agents, the models need to evolve to mimic the behaviors of agents. The evolved models show good match to the original agents in terms of local behaviors (parameter space) and global behaviors. The evolved classifiers perform collectively well and could be potentially used for detecting abnormal behaviors (e.g., faulty agents) in the swarm. We have also shown that swarm behaviors can be directly inferred from the motion of a single agent in the group, as long as the group size is sufficiently large. This may have significant implications for the study of animal collectives, as in practice it may be difficult to track a large number of animals in a group.

## 6 Conclusion

When inferring the deterministic/stochastic behaviors of a single agent, we extend the ability of the classifiers in *Turing Learning*. That is, the classifiers can interact with the agent through controlling the environmental stimulus that the agent responds to. The interactive learning approach proves to be superior to passive learning in terms of model converging rate and model accuracy, especially when the agent behavior under investigation has low observability. In the case study about inferring deterministic behaviors of a single agent, we have shown that it is possible to infer the behaviors through coevolving a fixed sequence of inputs and models. Therefore, in this case, the interaction between the classifiers and agent is not very ‘intelligent’ as they only need to output a fixed sequence no matter what the agent’s observed behavior is.

In a later case study, in which *Turing Learning* was applied to infer the stochastic behaviors of a single agent. We have shown that through actively interacting with the agent during the experimental process, the classifiers can ‘intelligently’ change the stimulus based on the agent’s observed behavior in order to extract the hidden information from the agent. In other words, the classifiers could dynamically capture the motion of the agent to adjust the environmental stimulus. This intelligent behavior is also observed in humans when scientists try to investigate the animals’ behavior in response to stimuli [96]. As the agent’s behavior under investigation is stochastic, it is impossible to evolve a fixed sequence of inputs to extract all the agent’s behavioral information. Given the same input, the agent would probably behave differently and this make it hard to optimize the models using predefined metrics. We compared the results obtained by *Turing Learning* and two other metric-based system identification methods, and show that *Turing Learning* learn the agent behavior significantly better than the other two methods. This highlight the benefits of *Turing Learning* in inferring agent behavior.

### 6.2 Future work

In spite of the encouraging results obtained when applying *Turing Learning* to infer agent behaviors, we do not claim that this method can be directly used for modeling animal behaviors. There are still questions need to be discussed before conducting experiments on animals. In the following, we provide some points that could be considered as our future work.

- In the thesis, the models are represented by a set of parameters that govern behavioral rules of the agents. As we argued before, this makes it feasible for us to

objectively gauge the quality of the models through comparing the ground truth. In the future, we will try to evolve the structure of the models as well (e.g., using genetic programming or artificial neural network).

- The swarm behaviors investigated in this thesis is deterministic in terms of the individual's behavioral rules. In the future, we could apply *Turing Learning* to learn such swarm behaviors that are stochastic. In fact, we have shown that if the original controller of the aggregation behavior investigated in Chapter 3 is stochastic, it is still possible to achieve the same global behavior [21]. For the stochastic aggregation controller, in each state, the agent has a probability ( $p$ ) of switching to another state (i.e., its binary sensor is flipped with certain probability). In this case, *Turing Learning* could be used for inferring both the controller and the probability.
- We could apply our approach to learn more complex behaviors (for example, when the agents have more states or rules). When the behaviors become more complex, instead of analyzing only the motion of individual agents, more information (such as number of the agent's neighbors or its internal states) may need to be provided to the classifiers.
- In principle, the *Turing Learning* method is applicable to infer human behavior. It could evolve models that aim to pass the Turing Test [149], at least with regards to some specific subset of human behavior. In this case, the classifiers could then act as Reverse Turing Tests, which could be applied in situations where a machine needs to distinguish human agents from artificial ones as done, for example, by the “Completely Automated Public Turing test to tell Computers and Humans Apart” system (CAPTCHA) [171], which is widely used for internet security purposes. Another exciting application of *Turing Learning* is reversing engineering the signature of human beings. Given a collection of signature from a human being, we could coevolve models which are computer programs that automatically generate signature and classifiers (e.g., neural network) that distinguish between the signature from human or that generated by models. Once the models are obtained, we can use the models to generate electronic signature similar to that of the human being.