Detecting change points

###

February 9, 2018

Abstract

In many setting agents are required to communicate with other agents whom their type they are unfamiliar with. By maintaining beliefs over a set of hypothetical behaviors, or types an agent can improve its interaction with other agents in order to achieve a common goal. Many work already investigated ways to correctly and efficiently calculate the relatively likelihood of an agent's type as well as the values of any bounded continuous parameters within types. Those work all assumed stationarity (i.e., types does not change during time). The assumption of stationarity, even-though suitable for number of realistic scenarios, is a very strong assumption which can not be guarantied in many real world settings. In this work however, we are investigating settings in which an agent can change its type during the mission to be accomplished. Allowing the different agents change their types add complexity to the planning problem as now an agent need to recognize that a change have occur in addition to figuring out what is the new type of the agent it is interact with. We are offering a new way, an improved version of the change point detection algorithm CHAMP, to track change points during the performances of a joint task. The proposed method is evaluated in detailed experiments...TO COMPLETE WHAT ARE OUR EXPERIMENTS.

1 Introduction

2 Preliminaries

This paper's terminology follows the one used in albrecht et al. ([2]).

2.1 Model

We consider a multi-agent model where agents interact each other in order to achieve a common goal. The process starts at time $t = 0$. At time t, each agent i receives a signal s_i^t and independently chooses an action a_i^t from some countable set of actions A_i. We do not put any limitations on s_i^t's structure and dynamic. This process continues indefinitely or until some termination criterion is satisfied (i.e., the goal achieved).

We will use $P(a_i^t|H_i^t, \theta_i, p)$ in order to denote the probability in which the action a_i^t is chosen where $H_i^t = (s_i^0, ..., s_i^t)$ is agent i's history of observations, θ_i is i's *type* and $p = (p_1, ..., p_n)$ is a vector of *continuous parameters* in θ_j. Each type's *continuous parameters* influence this type's abilities and therefore on its action choice (e.g., if one of those internal parameters sets that this agent can only see black objects then the probability for it to choose an action involving a white object is very low). Since this work mainly focus on detecting type changing points and since the work of albrecht et al has provided a method for reasoning about the values of any bounded continuous parameters within types, we will assume that the reasoning about the likelihood of a type also includes the evaluation of its internal continuous parameters. Therefore we will adjust the above notation to be: $P(a_i^t|H_i^t, \theta_i)$.

To simplify the exposition, we assume that we control a single agent, i, which reasons about the behavior of another agent, j. We also assume that i knows j's action space A_j and that it can observe j's past actions, i.e. $a_j^{t-1} \in s_i^t$ for $t > 0$. The true type of j, denoted θ_j^* is unknown to i. However, i has access to a finite set of *hypothetical types* $\theta_j \in \Theta_j$, with $\theta_j^* \in \Theta_j$. We furthermore assume that i has all information relevant to j's decision making, so that H_j^t is a function of H_i^t and we can write $P(a_j^t|H_i^t, \theta_j)$. Finally, we assume that agent j will change his type during the process in a number of chosen time points $\Lambda = \{\lambda_1, ..., \lambda_k\}$ (the criteria for choosing those time points will be elaborate later on the paper).

Our goal is to devise a method which allows agent i to be able both to identify the specific time point in which the change in agent j's type has occur and its new type, based only on agent j's observed actions.

3 Analysis

3.1 Changepoint Detection Method

First, we describe the CHAMP algorithm (**Ch**angepoint detection using **A**pproximate **M**odel **P**arameters) presented by Niekum et al. [4]. CHAMP is an algorithm for online Bayesian changepoint detection in settings where it is difficult or undesirable to integrate over the parameters of candidate models. The CHAMP algorithm was presented as an improvement of the online MAP (maximum a posteriori) changepoint detection model of Fearnhead and Liu [3]. In order to allow finding changepoint in cases where the parameters of the underlying model can not be marginalized CHAMP allows the use of models of any form, in which parameter estimates are available via means such as maximum likelihood fit, MCMC, or sample consensus methods. CHAMP requires three main changes from the work of Fearnhead and Liu: (i) *Approximate model evidence*- instead of calculating directly model evidence for each segment by integrating over the parameters of candidate models they used the Bayesian Information Criterion (BIC) in order to approximate it; (ii) *Minimum segment length*- since the model evidence was no longer well-defined a minimum segment length, α, was chosen for all models (i.e., changepoints can only begin to be considered at time $t = 2\alpha$); (iii) *Particle definition*- they revised the definition of a particle such that each particle also include the model relevant to each time step.

3.2 The TCD (Types Changepoint Detection) Algorithm

In our case, however, the parameters of the underlying model can be marginalized. This means that both the first and second adjustments required in CHAMP is not required in our case. On the other hand some additional improvement can be made in order to achieve better results and decrease calculation time and complexity:

- Use real likelihood instead of approximate likelihood. Up until now
- We correct the existing approach such that it will start each segment with random prior instead of prior that is being influenced by the results from the last segment (since the segment is not Dependant in each other).

TODO: DESCRIBED **TCD** AND GIVE A PSEUDO CODE FOR IT

4 Experimental Evaluation

We provide a detailed experimental evaluation of our **TCD** in the level-based foraging domain [1], which was introduced as a test domain for ad hoc teamwork [5].

- We will compare the following four cases:
  1. **Perfect Information -** whenever agent j changes its type it reveal its new type (i.e., agent i knows that agent j changed its type and also knows what the new type is).
  2. **No Changes Allowed -** agent i knows agent j's initial type but assumes that no changes of type are allowed/possible.
  3. **Stationary Types -** no change of type is being allowed, Agents have the same type during the whole process (Using Stefano's method in order to figure out j's type).
  4. **Dynamic Types -** agent j can change its type without revealing it to agent i which will use **TCD** in order to identify changepoints.

  We note that the Monte Carlo Tree Search (MCTS) planning algorithm was used in each time step in order to compute optimal actions with respect to the agent's beliefs and types.

5 Related work

cnclkcm

6 Conclusions

cneqioecoida

References

[1] Stefano V Albrecht and Subramanian Ramamoorthy. A game-theoretic model and best-response learning method for ad hoc coordination in multiagent systems. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1155–1156. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[2] Stefano V Albrecht and Peter Stone. Reasoning about hypothetical agent behaviours and their parameters. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 547–555. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

[3] Paul Fearnhead and Zhen Liu. On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):589–605, 2007.

[4] Scott Niekum, Sarah Osentoski, Christopher G Atkeson, and Andrew G Barto. Champ: Changepoint detection using approximate model parameters. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 2014.

[5] Peter Stone, Gal A Kaminka, Sarit Kraus, Jeffrey S Rosenschein, et al. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *AAAI*, 2010.