

生活パターン学習による行動予測エージェントシステム

田島 見祐 近藤 大輔 小川 均

立命館大学情報理工学部情報コミュニケーション学科

1 はじめに

近年、人の生活パターンも多様化し、各個人に対応するサービスを提供するための規則を詳細に定義することは困難となっている。そこで、各個人の生活パターンを発見できれば、各個人に対応する様々なサービスを提供することができる。本研究では、サービスを提供するための環境が整っているため、各個人の行動記録を収集することで、生活パターンを発見し、個人に対応するサービスを提供することを目的としている。また、このような環境を利用するため、なるべく少量のデータで早期に生活パターンを発見することが重要となる。

2 エージェントシステム

システムの実装には、Java 言語で実装されたエージェントプラットフォームである JADE を用いる。本システムでは、複数のエージェントが同一のエージェントプラットフォーム内に存在し、異なる端末で動作していることを想定している。

2.1 認証・記録エージェント

個人の認証および行動記録の収集は Felica を用いて行う。Felica には、製造 ID(IDm)と製造パラメータ(PMm)が定められており、これらの値で個体を識別している。本研究では、これら2つの値をデータベースに格納し、参照することで個人認証を行っている。対象者は認証後に GUI を操作し、自らの行動を示す。認証エージェントは ACL メッセージ[1]を用いて、個人の行動情報を記録エージェントに知らせる。記録エージェントは行動情報をデータベースに蓄積する。

2.2 推論エージェント

推論エージェントは、図 1, 2, 3 に示すようなクラス階層を保持している。これらのクラス階層に基づき、帰納的学習手法を用いて推論を行う。

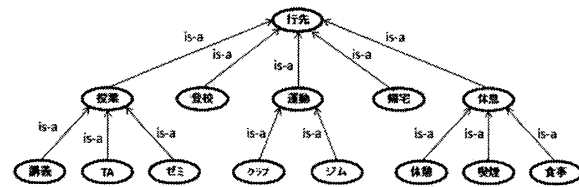


図 1: 行先のクラス階層

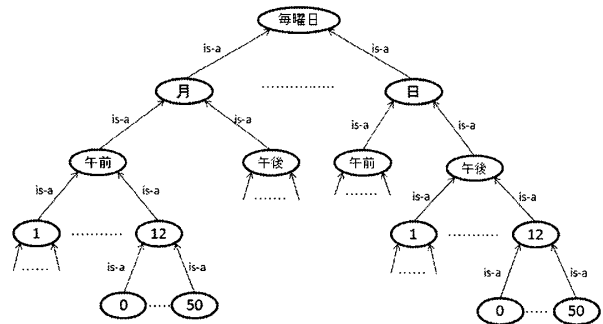


図 2: 曜日および時刻のクラス階層

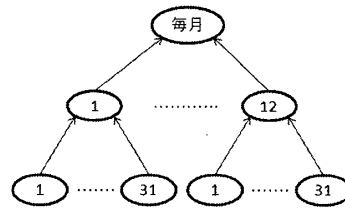


図 3: 月日のクラス階層

3 帰納的学習手法

本研究では、初期モデルと順に提示された例を解析することで、差異を発見し、行動モデルとして学習する手法を提案する。行動として正しい例を正例、正しくない例を負例、正例と類似している負例をニアミスと呼ぶ。初期モデルは正例でなくてはならない。推論には、正例とニアミスを用いる。正例が提示することでモデルの一般化を行い、ニアミスが提示することで特殊化を行う。特殊化では、モデルとニアミス間に重大な差異があるかが重要となる。学習を行うためには、より詳細なニアミスを提示する必要がある。本研究では、様々なニアミスを与え、推論された結果の評価を行う。

Behaviour Predicted by Learning Life Pattern of Agent Systems
 Kensuke Tashima (tashima@airlab.ics.ritsumeikan.ac.jp)
 Daisuke Kondoh (kondoh@airlab.ics.ritsumeikan.ac.jp)
 Hitoshi Ogawa (ogawa@airlab.ics.ritsumeikan.ac.jp)
 Information Science and Engineering, Ritsumeikan University

表 1: 実験データ

番号	行先	日にち	曜日	時刻
1	講義	11/2	月	12:00
2	ジム	11/3	火	12:00
3	講義	11/4	水	12:00
4	データなし	11/5	木	12:00
5	食事	11/6	金	12:00
6	講義	11/7	月	12:00
7	データなし	11/8	火	12:00
8	講義	11/9	水	12:00
9	食事	11/10	木	12:00
10	食事	11/11	金	12:00

4 実験

表 1 はある人物の正午の 2 週間の行動記録を表したものである。表 1 のデータを推論エージェントに与えた場合の推論について説明する。番号 1 の例を初期モデルとし、番号順に初期モデルと比較する。初期モデルは推論の結果により、適当な処理が行われ、更新されていく。以下、更新後の初期モデルをモデルという。まず、番号 2 の例と初期モデルの比較を行う。番号 2 の例はニアミスであり、初期モデルに存在しない頂点<ジム>、<火>を保持している。ここでは例としてわかりやすくするため、曜日についてのみ forbid-link 処理 [2]を行う。処理後のモデルを図 4 に示す。次に、番号 3 の例とモデルを比較する。差異である頂点<月>と<水>はクラス階層の一部であるため、climb-tree 処理 [2]を行う。処理後のモデルを図 5 に示す。図中の毎曜日とは、すべての曜日を表すものとする。番号 4, 5 の例の処理については、番号 2 とモデルの比較の場合と同様に forbid-link 処理 [2]を行う。処理後のモデルを図 6 に示す。ここで、処理後のモデルに上位クラスがあるにも関わらず、must-not 条件 [2]のついた枝を持つ頂点が複数存在しているため、これらに対して drop-link 処理 [2]を行い、その後、enlarge-set 処理 [2]を行う。1 週間目のデータを用いることで、行先と曜日および時刻の関係性を学習し、図 7 のモデルが導出することが可能である。

続いて、2 週目のデータを用いて、図 7 のモデルに日にちについての関係を追加する。元来、曜日と日にちには関係性があるため、この関係に基づき、推論を行う。ここでは、番号 6 と番号 8 の正例に着目する。<11/9>と<11/11>はクラス階層の一部であるため、climb-tree 処理 [2]を行う。この処理により、1 週目終了時に日にちが関連付けられ、図 8 の最終モデルが導き出される。

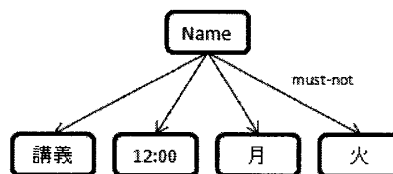


図 4: 番号 2 と比較後のモデル

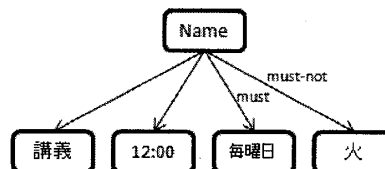


図 5: 番号 3 と比較後のモデル

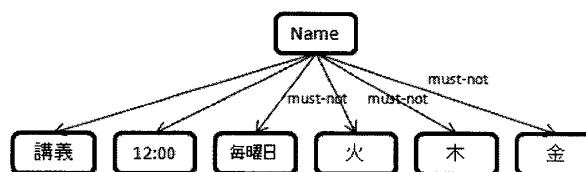


図 6: 番号 5 と比較後のモデル

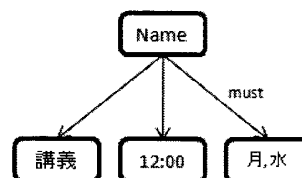


図 7: 1 週目終了後のモデル

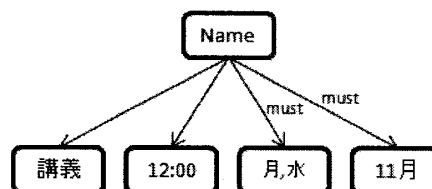


図 8: 最終モデル

5 おわりに

本稿では、帰納的学習手法を用いて、生活パターンを学習する機能を持つエージェントシステムの実装例を示した。今後はクラスを細分化し、より詳細な生活パターンを発見できるシステムを目指す。

参考文献

- [1] Fabio Luigi Bellifemine, Giovanni Caire, Dominc Greenwood, “developing multi-agent systems with JADE”, WILEY, 2007
- [2] 小川均, “知識工学”, 東京 共立出版, 2005.10