Hidden Markov Modeling of Team-Play Synchronization

Itsuki Noda[1,2]

[1] Cyber Assist Reseach Center
National Institute of Advanced Industrial Science and Technology
[2] PRESTO, Japan Science and Technology Corporation (JST)

Abstract. Imitation Learning is considered both as a method to acquire complex human and agent behaviors, and as a way to provide seeds for further learning. However, it is not clear what is a building block in imitation learning and what is the interface of blocks; therefore, it is difficult to apply imitation learning in a constructive way. This paper addresses agents' intentions as the building block that abstracts local situations of the agent and proposes a hierarchical hidden Markov model (HMM) in order to tackle this issue. The key of the proposed model is introduction of gate probabilities that restrict transition among agents' intentions according to others' intentions. Using these probabilities, the framework can control transitions flexibly among basic behaviors in a cooperative behavior. A learning method for the framework can be derived based on Baum-Welch's algorithm, which enables learning by observation of mentors' demonstration. Imitation learning by the proposed method can generalize behaviors from even one demonstration, because the mentors' behaviors are expressed as a distributed representation of a flow of likelihood in HMM.

1 Introduction

Imitation learning is considered to be a method to acquire complex human and agent behaviors and as a way to provide seeds for further learning [6, 9, 7]. While those studies have focused on imitating behaviors of single agents, few works address imitation for teamwork among multiple agents because the complexity of the world state increases drastically in multi-agent systems. On the other hand, stochastic models like hidden Markov models (HMM) have been studied as tools to model and to represent multi-agent/human interactions [8, 3, 2]. The merit of stochastic models is that we can apply the models in both behavior recognition and generation. However, it is hard to apply these stochastic models to imitate teamworks by observation because of the complexity of the model of multiple agents. This study focuses upon *intention*s of agents as building blocks of an abstract state of the local world for the agent in order to overcome the problem. Using *intention*, I formalize teamwork and propose a hierarchical hidden Markov model for imitation learning of teamwork.

D. Polani et al. (Eds.): RoboCup 2003, LNAI 3020, pp. 102–113, 2004.

2 Teamwork and Imitation

What is *teamwork* in multiagent systems? Consider a case wherein two soccer players pass a ball with each other. When a passer, who is keeping the ball, starts to pass the ball to a receiver, the passer must know that the teammate is ready to receive. Also, the receiver will start free-running when the receiver recognizes that the passer is looking for the receiver for a through-pass. This example illustrates that recognition of the others' intentions is important factor for decision making of player's intention. We enhance usage of intentions to derive teamwork. This section formalizes teamwork from intention and makes a correspondence to imitation learning.

2.1 Intention and Play

We suppose that an *intention* is a short-term idea to achieve a certain condition from another condition. For example, in soccer, the intention 'to guide a ball in a certain direction' is an idea to move to a certain direction with the ball. We assume that an *intention* is an individual idea; therefore, an agent does not pay attention to others' efforts to achieve the intention.

A *play*, as opposed to a *team-play*, is postulated as a sequence of atomic actions to achieve a single *intention*. The *play* is a basic building block of overall behavior of agents. For example, in soccer, a 'dribble' is a *play* to achieve the *intention* 'guide a ball in a certain direction', which consists of atomic actions like 'turn', 'dash', 'kick', and so on. A play for the intention is also an individual behavior without collaboration with other agents because an intention is an individual idea.

We also assume that a play corresponds to just one intention. Therefore, we use the word "play" and "intention" in the same meaning in the remainder of this article.

As shown below, an intention and the corresponding play are used as a main trigger to synchronize team-plays among multiple agents. This means that the intention is treated as a kind of partial condition of the world. For example, the fact that a player has a 'dribble' intention implies the following conditions: the player is keeping the ball; the player is moving toward a certain place; and the player may require teammates for support. In other words, an intention represents abstracted and simplified conditions of the world.

2.2 Team-Play

We suppose that *team-play* is a collection of plays performed by multiple agents to achieve a certain purpose. As mentioned in the previous section, an intention is an individual idea. This means that multiple agents who do not change their intentions can not perform a *team-play* because they have no way to synchronize their plays. Instead, we assume that they can synchronize their plays by changing their intentions according to situations of environments and intentions of other agents. For example, in soccer, when two players (passer and receiver) guide a ball by dribble and pass, players will change their intentions as shown in Fig. 2. In this example, the passer and the receiver initially have intentions 'dribbling' and 'supporting', respectively. Then, the passer changes the intention to 'seek-receiver', followed by the receiver's change to 'free-run', the passer's change to 'pass', and so on. Play synchronization is represented as conditions when agents can change the intention. In the example, the passer changes its intention from 'seek-receiver' to 'pass' when the teammate's intention is 'free-run'. In other words, we can denote the condition as follows:

Intent(passer,seek-receiver) & **Bel**(passer, **Intent**(receiver,free_run))
→ **Intent**(passer, pass)

2.3 Imitation Learning of Team-Play

Finally, we formalize imitation learning of the team-play.

In general, the imitation learning process is: (1) to **observe** behaviors of a mentor and interpret them into internal representation; (2) to **extract** rules of behaviors from internal representation; and (3) to **generate** a behavior based on rules. In the context of the *team-play* formalized in the previous section, the above process is realized as;

Observation phase: to observe behaviors of mentors and estimate what intention each agent has at each time step.
Extraction phase: to extract conditions prevailing when each agent changes intentions. A condition is represented as a conjunction of others' intentions.
Generation phase: to generate a sequence of intentions according to changes of environment and others' intentions.

In the **Extraction** phase of this process, the *intention* plays an important role: that is, conditions of changes of intentions. As described in Section 2.1, we consider that *intention* can represent world conditions. In addition to it, we use only *intention*s to construct rules for agents to change their *intention*. Although such abstraction reduces performance to express detailed conditions of the world, it provides good features for the machine learning. One important issue in machine learning is how to represent the state of a complex environment. This becomes a serious problem under a multi-agent environment because the number of factors to take into account increases exponentially in such environment. Abstraction of the world state by intentions can reduce the number of factors during the condition significantly. This kind of abstraction is necessary for imitation learning because only a small number of examples are given for imitation learning.

3 Hierarchical Hidden Markov Model for Agents

3.1 Basic Behavior Model

We formalize behaviors of a basic play m performed by a single agent as a Moore-type HMM as follows:

$$\mathbf{HMM^b}_m = \langle \boldsymbol{S}_m, \boldsymbol{V}_m, \boldsymbol{P}_m, \boldsymbol{Q}_m, \boldsymbol{R}_m \rangle \,,$$

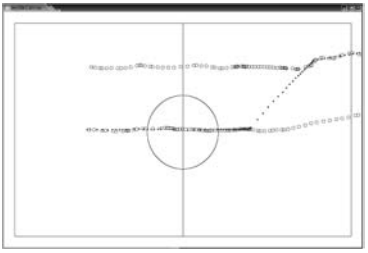

Fig. 1. Dribble and Pass Play

passer	receiver
dribbling	:
↓	supporting
seek-receiver	↓
↓	free-run
pass	↓
:	receive

Fig. 2. Changes of Intentions in Dribble and Pass Play

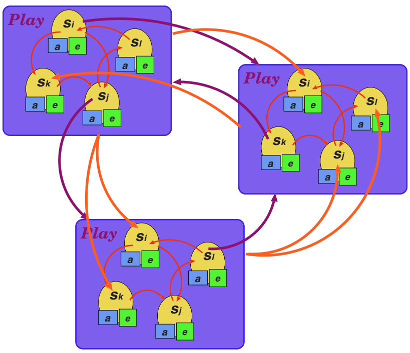

Fig. 3. Complex Behavior Model

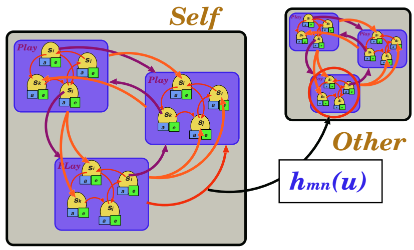

Fig. 4. Joint Behavior Model

where $\boldsymbol{S}_m$ is a set of states for the play m, $\boldsymbol{V}_m$ is a set of a pair of sensor value and action commands which are used as outputs from the state, $\boldsymbol{P}_m = \{p_{mij}\}$, $\boldsymbol{Q}_m = \{q_{mi}(v)\}$, and $\boldsymbol{R}_m = \{r_{mi}\}$ are probability matrixes of state transition, state-output, and initial state, respectively. These probabilities are defined as $p_{mij} = Pr(s_{mj}^{\langle t \rangle} | s_{mi}^{\langle t-1 \rangle})$, $q_{mi}(v) = Pr(v^{\langle t \rangle} | s_{mi}^{\langle t \rangle})$, and $r_{mi} = Pr(s_{mi}^{\langle 0 \rangle})$, where $s_{xyz}^{\langle t \rangle}$ means $s^{\langle t \rangle} = s_{xyz}$, and $\langle t \rangle$ on the right shoulder of a variable indicate the time t.

3.2 Complex Behavior Model

As discussed in the previous section, we consider that team-play consists of a sequence of intentions of multiple agents. This means that cooperative complex behavior of a single agent in a team of agents is considered as transitions among several basic plays ($\mathbf{HMM^b}$). Therefore, we formalize complex behavior as the following modified Mealy-type HMM (Figure 3),

$$\mathbf{HMM^c} = \langle \boldsymbol{M}, \boldsymbol{U}, \boldsymbol{E}, \boldsymbol{F}, \boldsymbol{G}, \boldsymbol{H} \rangle ,$$

where $\boldsymbol{M} = \{\mathbf{HMM^b}_m\}$ is a set of basic plays and $\boldsymbol{U}$ is a set of output from the model (normally, same as $\boldsymbol{M}$); $\boldsymbol{E} = \{e_m\}$ is a set of initial play probabilities, $\boldsymbol{F} = \{f_{min}\}$ is a set of exiting probabilities from plays, and $\boldsymbol{G} = \{g_{mnj}\}$ is a set of entering probabilities to plays. Also, $\boldsymbol{H} = \{h_{mn}(u)\}$ is a set of gate probabilities between plays. Formally, these probabilities are defined as: $e_m = Pr(\mathbf{HMM^b}_m^{\langle 0\rangle})$, $f_{min} = Pr(\mathbf{HMM^b}_n^{\langle t\rangle}|s_{mi}^{\langle t\rangle})$, $g_{mnj} = Pr(s_{nj}^{\langle t\rangle}|\mathbf{HMM^b}_m^{\langle t-1\rangle})$, and $h_{mn}(u) = Pr(u^{\langle t-1\rangle} \in \boldsymbol{U}|\mathbf{HMM^b}_m^{\langle t-1\rangle}, \mathbf{HMM^b}_n^{\langle t\rangle})$.

Using these probabilities, an actual probability from state i in play m to state j in play n is calculated as

$$p'_{minj} = Pr(s_{nj}^{\langle t\rangle}|s_{mi}^{\langle t-1\rangle}) = \begin{cases} f_{mim}p_{mij} & ; m = n \\ f_{min}g_{mnj} & ; m \neq n \end{cases} .$$

3.3 Joint-Behavior Model

Finally, we coupled multiple $\mathbf{HMM^c}$s , each of which represents the behavior of an agent. Coupling is represented by gate probabilities $\boldsymbol{H}$ (Fig. 4). For example, when agent X and agent Y are collaborating with each other, $h_{mn}(u)$ in $\mathbf{HMM^c}$ for agent X indicates the probability that agent Y is performing play u at time t when agent X changes the play from m to n during time $t \rightarrow t+1$.

3.4 Learning Procedure

Using the Baum-Welch algorithm, we can derive an learning procedure to adapt probabilities in $\mathbf{HMM^c}$. The first step is to calculate forward and backward likelihoods for each state and timestep in all plays as follows:

$$\begin{aligned} \alpha_{nj}^{\langle 0\rangle} &= e_n r_{nj} q_{nj}(v^{\langle 0\rangle}) \\ \alpha_{nj}^{\langle t+1\rangle} &= \left[\sum_i \tilde{f}_{nin}\alpha_{ni}^{\langle t\rangle}p_{nij} + \sum_{m\neq n} \bar{\alpha}_{mn}^{\langle t\rangle}g_{mnj}\right] q_{nj}(v^{\langle t+1\rangle}) \\ \beta_{mi}^{\langle T\rangle} &= 1 \\ \beta_{mi}^{\langle t-1\rangle} &= \left[\sum_j \tilde{f}_{mim}p_{mij}q_{mj}(v^{\langle t\rangle})\beta_{mj}^{\langle t\rangle} + \sum_{m\neq n} \tilde{f}_{min}\bar{\beta}_{mn}^{\langle t\rangle}\right] q_{nj}(v^{\langle t+1\rangle}), \end{aligned}$$

where

$$\begin{aligned} \bar{\alpha}_{mn}^{\langle t\rangle} &= \textstyle\sum_i \alpha_{mi}^{\langle t\rangle}\tilde{f}_{min} \\ \bar{\beta}_{mn}^{\langle t\rangle} &= \textstyle\sum_j g_{mnj}q_{nj}(v^{\langle t\rangle})\beta_{nj}^{\langle t\rangle} \end{aligned} \qquad \tilde{f}_{min} = \begin{cases} \frac{f_{min}+\lambda_{\text{sticky}}}{1+\lambda_{\text{sticky}}} & ; \text{if} m = n \\ \frac{f_{min}}{1+\lambda_{\text{sticky}}} & ; \text{otherwise} \end{cases} .$$

Here, λ_{sticky} is a positive value called a *sticky factor*. This factor is introduced because we should consider that an agent retains an intention relatively persistently. If the agent changes its intention repeatedly, it becomes difficult to estimate an agent's intention by observation, rendering complex behavior difficult. The sticky factor λ_{sticky} inhibits such frequent changes of intention during observation and estimation of mentors' behaviors. Note that the sticky factor is used only in the **Estimation** phase, and ignored in the **Generation** phase in the imitation learning.

Using α and β, we can adjust probabilities as follows:

$$e_m \leftarrow \sum_i \gamma_{mi}^{\langle 0 \rangle}, \quad f_{min} \leftarrow \frac{\sum_t \xi_{min}^{\langle t \rangle}}{\sum_t \gamma_{mi}^{\langle t \rangle}}, \quad g_{mnj} \leftarrow \frac{\sum_t \xi_{mnj}^{\langle t \rangle}}{\sum_t \gamma_{mn}^{\langle t \rangle}}, \quad h_{mn}(u) \leftarrow \frac{\sum_{t, u^{\langle t \rangle} = u} \gamma^{\langle t \rangle}}{\sum_t \gamma_{mn}^{\langle t \rangle}},$$

where

$$\begin{aligned} \xi_{min}^{\langle t \rangle} &= \alpha_{Mi}^{\langle t-1 \rangle} f_{min} h_{mn}(u^{\langle t-1 \rangle}) \bar{\beta}_{mn}^{\langle t \rangle} & \gamma_{mi}^{\langle t \rangle} &= \alpha_{mi}^{\langle t \rangle} \beta_{mi}^{\langle t \rangle} \\ \xi_{mnj}^{\langle t \rangle} &= \bar{\alpha}_{mn}^{\langle t-1 \rangle} g_{mnj} q_{nj}(v^{\langle t \rangle}) \beta_{nj}^{\langle t \rangle} & \gamma_{mn}^{\langle t \rangle} &= \bar{\alpha}_{mn}^{\langle t \rangle} h_{mn}(u^{\langle t \rangle}) \bar{\beta}_{mn}^{\langle t+1 \rangle}. \end{aligned}$$

3.5 Imitation Learning Using HMM

Using proposed HMMs, we realize the imitation learning as: (1) to train separately each $\mathbf{HMM^b}$ for each play; (2) to construct $\mathbf{HMM^c}$s using the trained $\mathbf{HMM^b}$s ; (3) to observe mentors' behaviors and environmental changes; then estimate likelihoods of each play (and each state in a play) at each time step by calculating forward and backward likelihoods ($\alpha_{nj}^{\langle t \rangle}$ and $\beta_{mi}^{\langle t \rangle}$) as shown in Section 3.4; (4) to adjust initial, exiting and entering probabilities ($\boldsymbol{E}$,$\boldsymbol{F}$ and $\boldsymbol{G}$) according to observation; (5) to repeat steps 3 and 4 until the probabilities converge (**Observation**); (6) to calculate gate probabilities ($\boldsymbol{H}$) using final forward/backward likelihood (**Extraction**); (7) to generate transitions among states and plays according to acquired probabilities (**Generation**). In the last phase, play generation is done as follows: The initial state for each agent is decided according to initial probability $Pr(s_{mi}^{\langle 0 \rangle}) = e_m r_{mi}$. When the play and the state of an agent at time t are m and s_{mi}, respectively and the set of others' plays is $u^{\langle t \rangle}$, then the next play n and the state i of the agent is decided according to the following likelihood L:

$$L(s_{nj}^{\langle t+1 \rangle}) = \begin{cases} f_{mim} p_{mij} q_{mj}(\hat{v}^{\langle t+1 \rangle}) & ; \ m = n \\ f_{min} g_{mnj} h_{mn}(u^{\langle t \rangle}) q_{nj}(\hat{v}^{\langle t+1 \rangle}) & ; \ m \neq n \end{cases} \tag{1}$$

where $\hat{v}^{\langle t+1 \rangle}$ is a partially observed output value in v at time $t+1$.

4 Experiments

4.1 Cyclic Alternating Shift Actions

Several simple experiments were conducted to demonstrate the performance of the proposed model.

In the experiments, two agents change four plays in a certain order by synchronizing them with each other. The four plays are: to move on a round path clockwise (A), to move on a round path counter-clockwise (B), to move in an '∞'-letter-shape path (C) , and to move in an '8'-letter-shape path (D). Actual

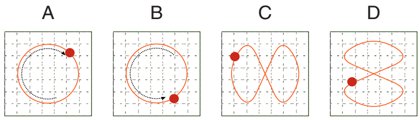

Fig. 5. Basic Plays used in Exp. 1 and Exp. 2

Exp. 1

Agent X	→A→B→C→D→A
Agent Y	D→C→B→A→D→

Exp. 2

Agent X	→A→B→C→D→A→D→C→B→A
Agent Y	A→B→D→A→C→D→B→C→A→

Fig. 6. Change Patterns of Team-plays used in Exp. 1 and Exp. 2

paths are shown in Fig. 5. The agents' actions (movement) are observed as a sequence of positions where the agent is located in each timestep[1].

We suppose that learner agents are already trained for these four plays; that is, the learners' $\mathbf{HMM^b}$s , each of which consists of 20 states, are already trained for these plays. This means that the $\mathbf{HMM^b}$s can generate required movements of corresponding play-A,-B,-C, and -D.

Note that only one example is given to the learner for each experiment in the following experiments, Because of imitation learning, learners should be able to generalize acquired behaviors from the example so that the learners can generate varied behavior according to difference of environments.

Exp. 1: Simple Shift of Plays: In the first experiment, each mentor agent simply changes its plays in a certain order (A →B →C →D for agent X and D →C →B →A for agent Y) alternately with each other as shown in Fig. 6-(a).

Figure 7 shows the relative likelihood of each play state for each agent at each timestep estimated by **Observation** phase. In this figure, there are eight rows of small squares: upper 4 rows correspond 4 plays of the first agent (agent X), and the rest 4 are plays for the second agent (agent Y). Each row corresponds to a play A, B, C or D in Fig. 5 respectively. In each row, a column consists of 20 small squares each of which corresponds a state of $\mathbf{HMM^b}$ for the play A–D at a certain timestep. The ratio of black area in the square indicates the relative likelihood with which the state of the $\mathbf{HMM^b}$ is active at the timestep. Columns are aligned along with time. So, a horizontal line of squares means changes of likelihood of a state of $\mathbf{HMM^b}$. From this figure, we can see that the learner estimate that the agent X behaves according to play-A at the beginning (states for play-A (squares in the most upper row) are active in the left most part of the figure), then adopts play-B, play-C, play-D continuously; it then returns to play-A, followed by the same changes. Similarly, the learner estimates that agent Y behaves play-D first, then changes plays in the reverse order of agent X. In addition to it, the change from play-A to play-B, from play-B to play-C, and from play-C to play-D in the agent X occur while the agent Y is doing play-C,

[1] Actually, the world is quantized into 49 (7×7) blocks when it is observed. Therefore, movements are observed as a sequence of blocks in which the agent is at each timestep.

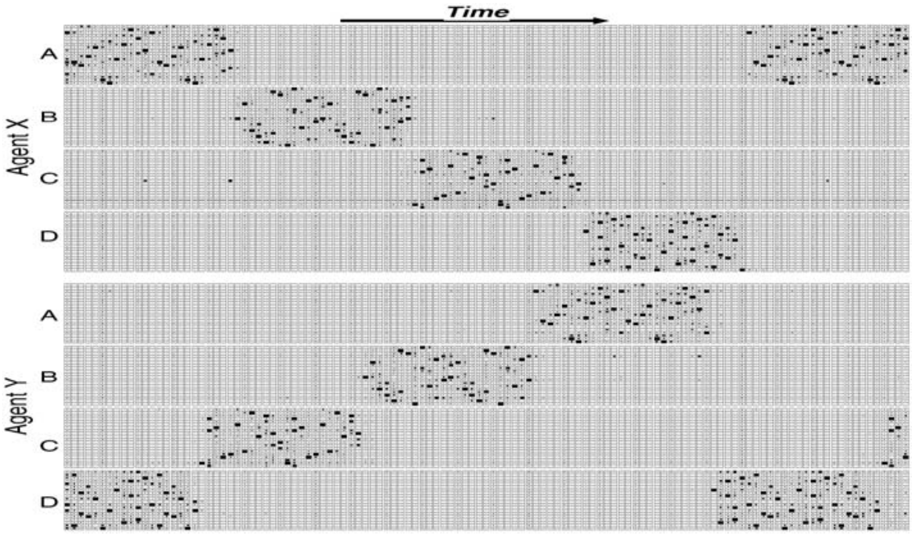

Fig. 7. Exp. 1: Result of Recognition of Mentors' Behaviors

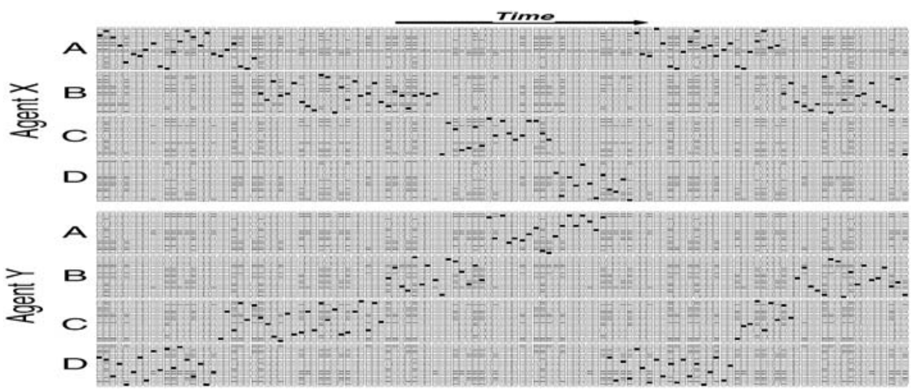

Fig. 8. Exp. 1: State Transitions Generated by Learned HMM

play-B, and play-A, respectively. These conditions are consistent with the change pattern of the mentor shown in Fig. 6.

Using the result of the estimation, the learner acquires probabilities in $\mathbf{HMM^c}$ as conditions of changes of plays. For example, probabilities to transit from play A to play B and from play B to play C in the agent X were acquired in a certain trial as follows[2]:

$$\begin{array}{lll} h_{AB}(C) = 1.00, & f_{A8B} = 0.33, & f_{A14B} = 1.00, \\ h_{BC}(B) = 1.00, & f_{B3C} = 0.71, & f_{B6C} = 0.96. \end{array} \quad (2)$$

[2] Actual values of these probabilities vary according to initial values before learning and random noise added in the mentors' behaviors.

These probabilities represent the following conditions about changes of plays:

Intent(X,A) & **Bel**(X, **Intent**(Y, C)) → **Intent**(X, B)
Intent(X,B) & **Bel**(X, **Intent**(Y, B)) → **Intent**(X, C)

Using these probabilities, the learner can generate similar behaviors to those shown in Fig. 8. This figure is constructed in the same way as Fig. 7, but only one square is filled in a timestep because the learner decides one of the possible states according to the likelihood shown in Eq. 1. From this figure, we can see that the learner generates behaviors in the same manner of the mentor; that is, the order of the generated plays of agent X and Y are 'A →B →C →D' and 'D →C →B →A' respectively. Also, timings of the change of plays are consistent with the mentor's demonstration.

Generated behaviors are not deterministic because the acquired probabilities may take intermediate values as like f_{A8B} and f_{B3C} in Eq. 2. For example, durations of the play-C in agent Y are different in the first cycle and the second cycle in Fig. 8. This means that the learner has the ability to adapt to difference of the environment using methods for HMM such as Vitabi's algorithm.

Exp. 2: Conditional Change of Plays: The second experiment illustrates that the proposed framework can learn conditional transitions of plays using change pattern shown in Fig. 6-(c). The change pattern of Fig. 6-(c) includes conditional branching of the transition of plays. For example, agent X may change its play from A to two possible destination, B or D. The change can be decided encountering agent Y's play. When agent Y is doing play-B, agent X changes its play from A only to B; when agent Y is play-D, agent X changes to D. Figure 10 shows resultant behaviors generated after learning. As shown in this figure, the learner acquires correct conditions of the branching transition. For example, changes from play-A to -B of agent X only occur during agent Y's play-B. Actually, these conditions are represented by gate probabilities: for example, $h_{AB}(B) = 0.97$ and $h_{AD}(B) = 0.00$ for agent X.

4.2 Exp. 3: Dribble and Pass Play in Soccer

Finally, I applied the framework to collaborative play of soccer. The demonstration by mentors is dribble and pass play as shown in Fig. 1: A player starts to dribble from the center of the left half field and brings the ball to the right half. At the same time, another player runs parallel along the upper (or lower) side of the field supporting the dribbling player. Then, the first player slows to look-up the second player; it then passes the ball to that player. Simultaneously, the second player starts to dash to the ball and dribbles after receiving the ball. After the pass, the first player exchanges roles with the teammate so that it becomes a supporting player for the second player.

To imitate this demonstration, I trained six $\mathbf{HMM^b}$s to model 'dribble', 'slow-down and look-up', 'pass', 'free-run', 'chase-ball', and 'support'. Each of $\mathbf{HMM^b}$ has 5 states. The output of these $\mathbf{HMM^b}$s consists of local situations

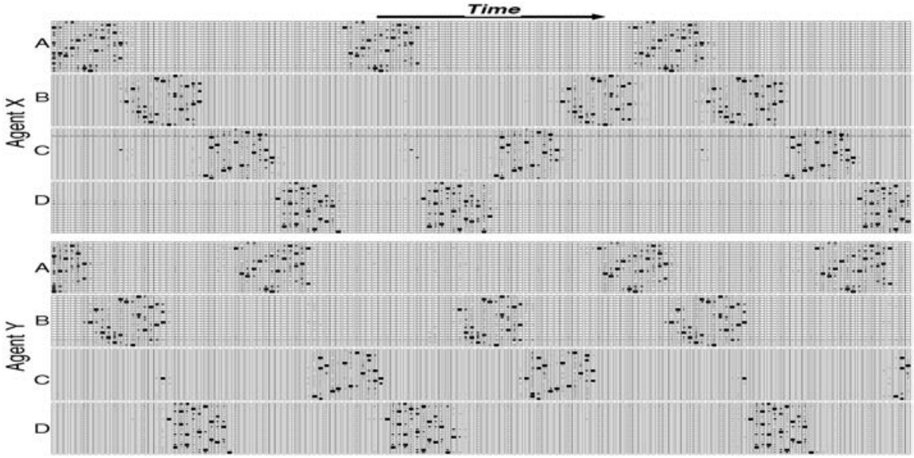

Fig. 9. Exp. 2: Result of Recognition of Mentor's Behaviors

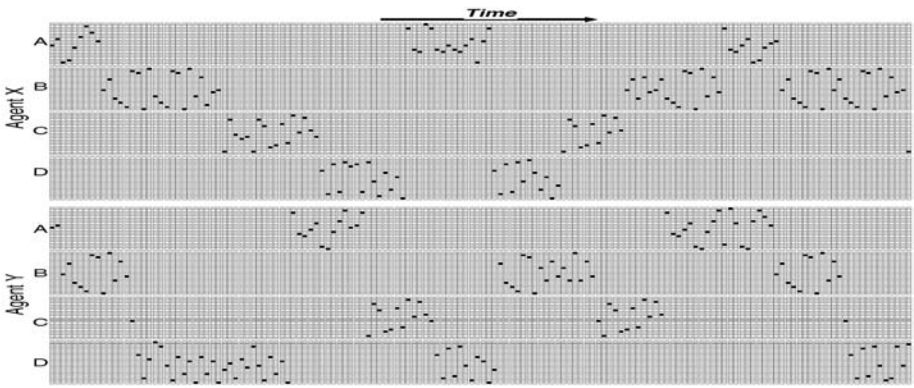

Fig. 10. Exp. 2: State Transitions Generated by Learned HMM

(the relative position and the velocity to the ball) and agent's actions ('turn', 'dash', 'small-kick', 'long-kick', 'trap', and 'look'). Note that there is no information about others' situations for output of $\mathbf{HMM^b}$s . As described in Section 2.2, others' situations are taken into account during the **Extraction** phase in learning.

Two $\mathbf{HMM^c}$s for agent X (the first player) and Y (the second player) are constructed after the training of the $\mathbf{HMM^b}$s . Then, the learner observes behaviors of the mentor and adjusts probabilities of the $\mathbf{HMM^c}$s .

Figure 1 shows result of observation and estimation. There are six $\mathbf{HMM^b}$s in this experiments; therefore, there are six rows (denoted by D, K, P, F, C, and S) for each agent, in which a column consists of five squares. Rows mean 'dribble (D)' , 'slow-down and look-up (K)', 'pass (P)', 'free-run (F)', 'chase-ball (C)', and 'support (S)', respectively. From this figure, we can see that the learner estimates changes of play for agent X and Y are 'dribble' →'slow-down' →'pass'

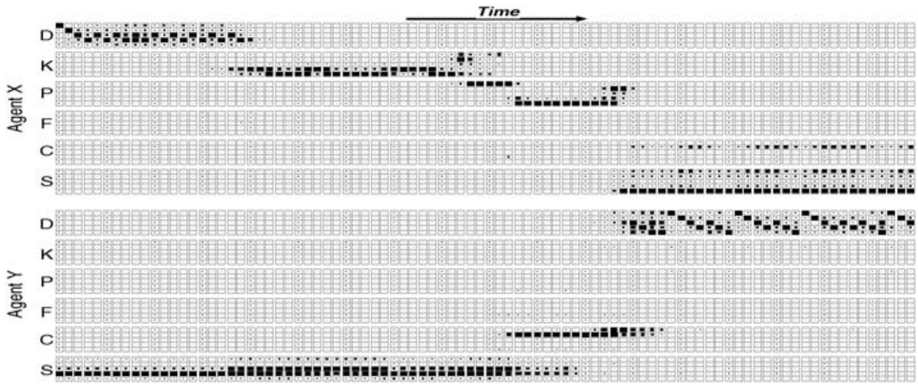

Fig. 11. Exp. 3: Result of Recognition of a Mentor's Behaviors

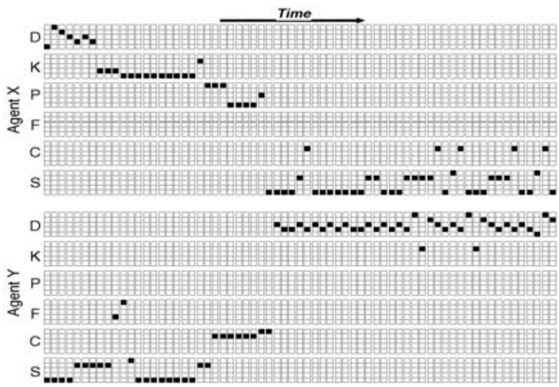

Fig. 12. Exp. 3: State Transitions Generated by Learned HMM

→'support' and 'support' →'chase-ball' →'dribble'. Consequently, the learner can generate various behaviors similar to the demonstration as shown in Fig. 12. In this example, although the learner sometimes generates wrong state transitions, for example a transition to states to the 'free-run' play in agent Y during agent X is doing 'slow-down', it recovers to the suitable transitions and continues to imitate the demonstrator. This shows robustness of the model against accidents. Because the model is coupled loosely with world and other's states by output probabilities of HMM, it can permit variation and misunderstanding of world and others' states.

5 Related Works and Discussion

There are several works on coupling HMMs that can represent combinational probabilistic phenomena like multi-agent collaboration [5, 1, 4]. In these works, probabilistic relation among several HMMs (agents) are represented as state-transition probabilities, such that the amount of memory complexity increases exponentially. This is a serious problem for imitation learning because we assume that the number of examples is small for imitation. In our model, the relation among agents is represented by gate probabilities $\boldsymbol{H}$, in which others' states are treated as outputs instead of as conditions of state transition. Using them, the likelihoods of state-transitions are simplified as products of several probabilities (Eq. 1). In addition, detailed states of other agents are abstracted by play (intention). As a result, the number of parameters is reduced drastically, so that learning requires very small number of examples as shown in above examples. Although such simplification may decrease flexibility of representation as a probabilistic model, experiments show that the proposed model has enough power to represent team-play among agents.

Intention in the model brings another aspect to communication among agents. We assume that there are no mutual communication in the proposed model. However, we can introduce communication as a bypass of observation and estimation of other's intention (play). The proposed model will be able to provide criteria for when an agent should inform their intention to others by comparing agents' actual intentions and estimated intention of the agent itself by simulating its own $\mathbf{HMM^c}$.

One important issue is the design of the intention. In the proposed model, intentions play various important roles like chanking of the actions and conditions of world state. Therefore, we must design intentions carefully so that team-plays can be represented flexibly.

References

1. Zoubin Ghahramani and Michael I. Jordan. Factorial hidden markov models. *Machine Learning*, 29:245–275, 1997.
2. Stephen S. Intille and Aaron F. Bobick. Recognizing planned, multiperson action. *Computer Vision and Image Understanding: CVIU*, 81(3):414–445, 2001.
3. Y. A. Ivanov and A. F. Bobick. Recognition of multi-agent interaction in video surveillance. In *International Conference on Computer Vision (Volume 1)*, pages 169–176. IEEE, Sep. 1999.
4. Michael I. Jordan, Zoubin Ghahramani, Tommi Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.
5. Michael I. Jordan, Zoubin Ghahramani, and Lawrence K. Saul. Hidden markov decision trees. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, page 501. The MIT Press, 1997.
6. Y. Kuniyoshi and H. Inoue. Qualitative recognition of ongoing human action sequences. In *Proc. IJCAI93*, pages 1600–1609, 1993.
7. Hiroyuki Miyamoto and Mitsuo Kawato. A tennis serve and upswing learning robot based on bi-directional theory. *Neural Networks*, 11:1331–1344, 1998.
8. Nuria M. Oliver, Barbara Rosario, and Alex Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.
9. Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, Jun. 1999.