2005－UBI－9 (7)
2005／11／1

信用管理における確率モデルに基づく利用者プレゼンスの推定

高田 喜朗[†,††] 野田 潤[††] 高橋 三恵[††] 細見 格[††] 関 浩之[†,††]

† 奈良先端科学技術大学院大学 情報科学研究科 630-0192 奈良県生駒市高山町 8916-5
†† NEC インターネットシステム研究所 630-0101 奈良県生駒市高山町 8916-47
E-mail: †{y-takata,seki}@is.naist.jp, ††{j-noda@cw,m-takahashi@ha,i-hosomi@ay}.jp.nec.com

あらまし ユビキタス環境のための信用管理アーキテクチャの一部として，拡張された隠れマルコフモデルに基づく利用者プレゼンスの推定法を提案する．提案するプレゼンス推定エンジンは，不完全なセンサ情報を補うとともに，利用者プレゼンスの信頼度を信用管理エンジンに提供する．より正確な推定を行うため，センサ情報だけでなく，時間帯や利用者の予定表を観測可能変数として用いる．また，一つの場所に留まる時間の分布を表すため，ミクロ状態を導入する．評価実験により，提案手法がセンサシステムの出力の精度を改善することを示す．

キーワード 利用者プレゼンス, 信用管理, 隠れマルコフモデル, ユビキタス, 情報セキュリティ

User Presence Estimation Based on a Stochastic Model in Presence-aware Trust Management

Yoshiaki TAKATA[†,††], Jun NODA[††], Mie TAKAHASHI[††], Itaru HOSOMI[††], and Hiroyuki SEKI[†,††]

† Nara Institute of Science and Technology Takayama 8916-5, Ikoma, Nara, 630-0192 Japan
†† Internet Systems Research Laboratories, NEC Corporation Takayama 8916-47, Ikoma, Nara, 630-0101 Japan
E-mail: †{y-takata,seki}@is.naist.jp, ††{j-noda@cw,m-takahashi@ha,i-hosomi@ay}.jp.nec.com

Abstract As a part of a trust management architecture in ubiquitous environments, we propose an inference method for user presence based on an extended version of the hidden Markov model (HMM). The presence inference engine based on HMM complements incomplete sensor signals and provides a trust management engine with user presence and its confidence level. To provide a more refined stochastic model, we also use time intervals and user schedules as observable variables, and we introduce micro states to represent time durations in a certain place. The experimental results show that the proposed method improves the precision of user presence detected by the sensor system.

Key words User Presence, Trust Management, Hidden Markov Model, Ubiquitous, Information Security

1. Introduction

Trust management provides not only a user authentication method based on public-key infrastructure (PKI) instead of password authentication, but it also offers a flexible and systematic method for assigning roles, capabilities, or authorizations to a user the system doesn't recognize. Specifically, context information, including user presence, is crucial for the system when deciding whether to permit a user access to the system's resources, and what kind of service the system should provide for the user.

In this paper, we propose an inference method for user presence based on the hidden Markov model (HMM). The presence inference engine complements incomplete sensor signals and provides a trust management engine with user presence and its confidence level (i.e. the probability that the user may be in that position). In this architecture, we can make a trust management policy depending on the confidence level of user presence; e.g., user A can access the LAN when with a certain confidence A seems to be with a full-time employee in the same room.

To provide a more refined stochastic model, we use not only sensor signals but also time intervals and user schedules as observable variables, and we introduce micro states to represent time duration in a certain place. We numerically evaluated the prediction power of the presence inference engine using human subjects with RFID tags. The proposed method resulted in better precision than a few methods including the sensor subsystem itself and HMM without micro states.

Based on the proposed method of presence estimation, we implemented a prototype system that performs access

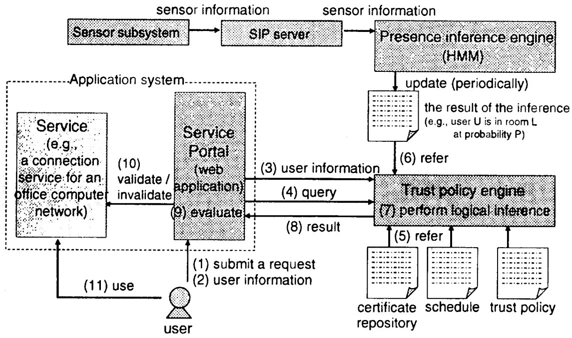

Fig. 1 Architecture of trust management system

controls for connection services in an office computer network [5] (Fig. 1).

Related work Inferences of user presence based on Bayesian models have been studied. SmartMoveX [3] is a system based on HMM. The Bayesian filter [1] is a general temporal model that includes HMM, and in which the state space can be continuous. Our model is an HMM augmented with extra variables and is not an instance of Bayesian filters but an instance of the dynamic Bayesian networks (DBN). Patterson et al. [4] used a DBN for GPS-based presence inference. Since they were motivated by an outdoor application, their model is quite different from ours.

CSAC [2] is an approach similar to ours; a service provider determines whether a service request is accepted based on information given by a context provider. They implemented a service portal for train travelers. Our model is more general than CSAC in that context parameters are freely introduced in a policy since our policy language is based on first-order logic.

2. Presence Inference Engine

The basic design of our stochastic model is as follows. User's (real) presence at a time instance is a random variable over a given set of positions (e.g. {office, meeting_room, cafeteria, home}). While the real presence is not observable, information that correlates to the presence (e.g. sensor signals, user's schedule, and current time) is observable. Moreover, the real presence changes as time goes on, and the current presence correlates to the previous presence. These correlations, depicted by Fig. 2, form a kind of the hidden Markov model (HMM). Below, we briefly review the definition of HMM, and present an extension of HMM for inferring a user's presence using time intervals and the user's schedule as well as sensor signals.

2.1 Hidden Markov model (HMM)

HMM is a stochastic model that has a similar structure to finite state machines. Formally, HMM is a 5-tuple $M = (S, T, \delta, \phi, I)$ where:

- S is a finite set of *states*.
- T is a finite set of *output symbols*.
- $\delta : S \times S \to \mathcal{R}$, where $\mathcal{R}$ is the set of real numbers, is *state transition probability distribution*.
- $\phi : S \times T \to \mathcal{R}$ is *output probability distribution*.
- $I : S \to \mathcal{R}$ is *initial state distribution*.

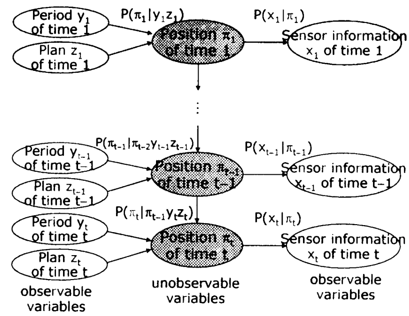

Fig. 2 Probability model for presence inference

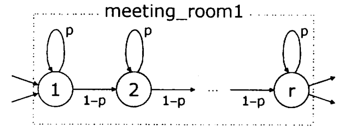

Fig. 3 Micro states

We sometimes use the random variable π_t that denotes the state of the model at a time instance t.

Posterior probability $p_s(t) = \Pr(\pi_t = s \mid x_1 \dots x_t)$ is the probability that the model is in state s at time instance t given that a sequence $x_1 x_2 \dots x_t \in T^*$ (where T^* is the Kleene closure of T) is observed. The probability $p_s(t)$ can be computed using the following equations.

$$f_s(t) = \Pr(x_1 \dots x_t,\ \pi_t = s) = \begin{cases} \phi(s, x_t) \sum_{s' \in S} f_{s'}(t-1)\delta(s', s) & t \geqq 1, \\ I(s) & t = 0, \end{cases}$$

$$p_s(t) = f_s(t) / \Pr(x_1 \dots x_t) = f_s(t) / \Big(\sum_{s' \in S} f_{s'}(t) \Big).$$

2.2 Duration distribution and micro states

In the above-mentioned definition, the length of a stay in the same state follows a geometric distribution. However, when we use HMM to infer a user's presence, it may be unnatural to assume that the probability that will user stay in a room (e.g. a meeting room) for 3 minutes is greater than the probability that he or she will stay there for one hour.

The use of micro states is a simple approach for representing a hill-shaped duration distribution. For each position l, we assign a subset of states to l, and consider that a user is in l whenever the model is in any member of the subset (Fig. 3). We call the subset and each member of the subset a *macro state* and a *micro state*, respectively. A macro state is defined by two parameters: the number r of micro states and the probability p of the self loop in every micro state. The probability $\Pr(d_l = k)$ that the model stays in the macro state l for k time instances follows a negative binomial distribution.

2.3 State transition probability distribution for presence inference

As shown in Fig. 2, the state π_t at time t is decided based on the state π_{t-1} at the previous time instance as well as the current period y_t and the scheduled plan z_t. Thus, the state transition probability distribution should be given by the conditional probability $\Pr(\pi_t \mid \pi_{t-1}, y_t, z_t)$ for each quadruple $(\pi_t, \pi_{t-1}, y_t, z_t)$. However, the size of the domain of the quadruple is very large for practical applications. Instead, we introduce a method for compounding independently defined probabilities $\Pr(\pi_t \mid \pi_{t-1})$, $\Pr(\pi_t \mid y_t)$, and $\Pr(\pi_t \mid z_t)$ into one probability $\Pr(\pi_t \mid \pi_{t-1}, y_t, z_t)$ using the following equation, where α and β are given weights.

$$\begin{aligned}\Pr(\pi_t \mid \pi_{t-1}, y_t, z_t) &= \alpha \Pr(\pi_t \mid \pi_{t-1}) + \beta \Pr(\pi_t \mid y_t) \\ &\quad + (1 - \alpha - \beta) \Pr(\pi_t \mid z_t), \\ 0 \leqq \alpha,\ 0 \leqq \beta,\ \alpha + \beta \leqq 1. &\end{aligned}$$

3. Experiments

We conducted a number of experiments to investigate the precision of the output of the presence inference engine. The following two sets of observation sequences were recorded and used as input for the presence inference engine. The former is concerned with investigating precision in detail while the latter is concerned with investigation in a realistic setting.

Experiment 1. The sequence of sensor signals observed when a subject was moving around in a building for about 30 minutes according to a predefined scenario.

Experiment 2. The sequence of sensor signals observed when a subject performed his/her ordinary work for a single day.

3.1 Measure of precision

Let N be the length of an observation sequence input into the presence inference engine. Let $sc = s_1 s_2 \ldots s_N$ be the sequence of correct presences. We define the *precision* of the output of the inference engine as $\left(\sum_{t=1}^{N} p_{s_t}(t)\right)/N$. Intuitively, the numerator represents the expected value of the number of correct answers if we assume that the inference engine chooses an arbitrary position s with probability $p_s(t)$ for each time t. Divided by the length of the sequence, precision is normalized within a range of between 0 and 1.

3.2 Sensor system

The presence inference engine used in the experiments works with an external sensor system using RFID tags. Each subject is carrying an active RFID tag while moving through a building. There are seven fixed sensors, roughly one per room, on a single floor of the building. Each sensor receives signals emitted by the RFID tags. The control computer (or controller) of the sensor system replies with the ID of the sensor that is receiving the signal from each tag. When more than one sensor is receiving the signal, the controller answers the one that is receiving the strongest radio wave signal. When there are no sensors receiving the signal, the controller answers "no signal."

3.3 Design of HMM

The set of HMM output symbols are defined according to the sensor system, i.e., it consists of the seven positions (or rooms) where RFID sensors are placed and one special symbol 'no_signal.' The ranges of the other random variables are defined as follows.

- positions (the range of π_t):
  - Seven positions where the sensors are placed: room_301, room_302, refresh_corner, lounge, office, east_office, laboratory.
  - Two positions where no sensor exists: hall, cafeteria.
  - Three virtual positions: another_place_in_building, out_of_building, home. The latter two positions are used only in Experiment 2.
- Periods (the range of y_t): AM, lunch, PM, off.
- Plans (the range of z_t): meeting_at_room_301, meeting_at_room_302, meeting_at_hall, no_plan.

After a one-hour discussion among a few of the authors, we defined the HMM parameters $\Pr(\pi_t \mid \pi_{t-1})$, $\Pr(\pi_t \mid y_t)$, $\Pr(\pi_t \mid z_t)$, and the mean μ_l and the standard deviation σ_l of the duration distribution so that the parameters would represent the behavior of an ordinary subject in the building.

The output probability $\Pr(x_t \mid \pi_t)$ was also defined as follows. For each room l where a sensor exists, we gave a high probability (more than 0.9) to $\Pr(x_t = l \mid \pi_t = l)$ and a low probability (less than or equal to 0.02) to any other x_t including 'no_signal.' However, room_301 and room_302 were adjacent and the probability that the sensor system would answer the room opposite to the correct one was not marginal. Hence, we gave a relatively high probability (0.13) to $\Pr(x_t = \text{room\_301} \mid \pi_t = \text{room\_302})$ and $\Pr(x_t = \text{room\_302} \mid \pi_t = \text{room\_301})$. We gave the same treatment for the office and east_office. For each place l where no sensor exists, we gave a high probability to $\Pr(x_t = \text{no\_signal} \mid \pi_t = l)$ and a low probability to any other x_t.

The weights α and β in Sect. 2.3 are defined as $\alpha = 1/(1 + r_{\beta,t} + r_{\gamma,t})$ and $\beta = r_{\beta,t}/(1 + r_{\beta,t} + r_{\gamma,t})$ where $r_{\beta,t}$ and $r_{\gamma,t}$ are ratios defined in the following way, i.e., we regard the current period and the current plan as important when it is lunch time or when some plan is specified.

$$r_{\beta,t} = \begin{cases} 10 \text{ if } y_t = \text{lunch} \\ 1 \ \text{ if } y_t = \text{off} \\ 0 \ \text{ otherwise.} \end{cases} \qquad r_{\gamma,t} = \begin{cases} 10 \text{ if } z_t \neq \text{no\_plan} \\ 0 \ \text{ if } z_t = \text{no\_plan.} \end{cases}$$

3.4 Experiment 1

3.4.1 Settings

We defined a scenario in which each subject moved through several locations including the office, meeting room, hall, and cafeteria within about 38 minutes. According to the scenario, six subjects (named a to f) were moving together. The output of the sensor system was recorded every six seconds (0.1 minute). The length N of the input sequence for each subject was 378.

Because the time spent at each position in the scenario was much shorter than in ordinary behavior, we defined the duration distribution according to the scenario.

We defined the following two different schedules and performed an analysis of precision for each schedule.

- *exact*: Each meeting is specified as taking as long as the correct one.
- *no_plan*: No plan is specified in this schedule.

3.4.2 Result

For each above-mentioned schedule, we compared the precision of the following four inference methods:

- *proposed*: the output of the proposed HMM.
- *no microstate*: the output of the proposed HMM without micro states.
- *sensor+schedule*: the output of the sensor system replaced the output symbol 'no_signal' according to the following rules.
  - If the current position l is specified in the schedule, then replace 'no_signal' with l.
  - Otherwise, replace 'no_signal' with 'another_place_in_building.'

We assumed that the posterior probability was 1 for the output position and 0 for the other positions.

- *sensor*: the same as *sensor+schedule* except that 'no_signal' was simply replaced with 'another_place_in_building.'

The results are summarized in Table 1. In each schedule, the proposed method resulted in better precision than the other methods used. In particular, when no plans were specified in the schedule, the proposed method gave much better precision than both the *sensor* and *sensor+schedule* methods. This suggests that our method is robust against incorrect schedules, which frequently appear in practice.

3.5 Experiment 2

On a single day, two subjects (the same as the subjects c and d in Experiment 1) performed their ordinary work while

Table 1 Result of Experiment 1

method	average
exact schedule	
proposed	0.786
no microstate	0.783
sensor+schedule	0.772
sensor	0.622
no-plan schedule	
proposed	0.713
no microstate	0.685
sensor+schedule	0.622
sensor	0.622

Table 2 Result of Experiment 2

	subject	
	c	d
proposed	0.788	0.842
sensor+schedule	0.723	0.796
sensor	0.723	0.796

the output of the sensor system was recorded. Moreover, the correct position at each time was recorded by each subject in a notebook. The output of the sensor system was recorded every minute, from morning till the subjects left the building (approximately eleven hours). The length N of the input sequence was 700 and 642 for subjects c and d, respectively.

The results are summarized in Table 2. The proposed method resulted in better precision than *sensor* and *sensor+schedule*. In this experiment, the precisions of *sensor* and *sensor+schedule* were the same, because there were no time instances during the experiment in which a plan was specified and 'no_signal' was observed.

4. Conclusion

We proposed a method for user presence estimation for presence-aware trust management in ubiquitous environments. Experimental results show that the proposed method outperforms a few methods including the sensor subsystem itself and HMM without micro states.

References

[1] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, Bayesian filtering for location estimation, IEEE Pervasive Computing, 2(3), 24–33, 2003.

[2] R. J. Hulsebosch, A. H. Salden, M. S. Bargh, P. W. G. Ebben and J. Reitsma, Context sensitive access control, 10th ACM Symp. on Access Control Models and Technologies (SACMAT), 111–119, June 2005.

[3] J. Krumm, L. Williams, and G. Smith, SmartMoveX on a graph — an inexpensive active badge tracker, UbiComp 2002, LNCS 2498, 299–307, 2002.

[4] D. J. Patterson, L. Liao, D. Fox, and H. Kautz, Inferring high-level behavior from low-level sensors, UbiComp 2003, LNCS 2864, 73–89, 2003.

[5] Y. Takata, J. Noda, M. Takahashi, H. Mouri, I. Hosomi, D. Taguchi and H. Seki, A presence-aware trust management system, Technical Report, Nara Institute of Science and Technology, Oct. 2005.