Multiple Sequential Data Modeling with Extended Hidden Semi-Markov Models and Its Data Aggregation and Management

Hiromi Narimatsu
Graduate School of Information Systems
The University of Electro-Communications

A thesis submitted for the degree of
Doctor of Engineering

December 2017

Multiple Sequential Data Modeling with Extended Hidden Semi-Markov Models and Its Data Aggregation and Management

Supervisory Committee

Chairman: Assoc. Prof. Hiroyuki Kasai
Member: Prof. Hiroyoshi Morita
Assoc. Prof. Tomohiro Ogawa
Assoc. Prof. Satoshi Ohzahata
Prof. Maomi Ueno

拡張隠れセミマルコフモデルによる複数系列データモデリングとデータ収集・管理手法

成松 宏美

概要

近年，デバイスの発展とデータ収集方式の発展に伴い，解析の対象となるデータや収集方式に変化が出来ている．例えば，農業向けのサービスでは．農地に温度や湿度センサをおいて，センサ値をクラウドサーバに送ることで，農地の状態遠隔監視や水量制御の自動調整を行っている．ヘルスケア向けのサービスでは，スマートフォンとスマートウォッチやフィットネスデバイス等のウェアラブルデバイス等で，所有者の状態を計測，管理している．同様のサービスは多々あるが，システム構成及びデータの特性として共通していることがある．1つ目は，センサ等の複数のデバイスのデータを1つまたは複数のアクセスポイント相当の端末を経由してクラウドに送信され，管理されること．2つ目は，1つの端末に収集されたセンシングデータは，1つのグループとして意味をなす集合であることである．農地のセンシングで言えば，中継機がセンサからデータを収集し，一定期間毎にクラウドサーバへデータをアップロードする．また中継機の収集したデータは，同一農地内のデータであるという点でグループとしても意味のあるデータである．ヘルスケアにおいても，スマートフォンがウェアラブルデバイスのデータを収集し，その収集されたデータは，同一所有者のデータであるという点でグループとしても意味のあるデータである．

しかしながらデータの解析，収集の観点で次のような2つの課題がある．1つ目は，複数の系列データをグループとして捉えた解析はあまり想定されていなかったこと．時事刻々と収集されるデータは”系列”という特徴があり，系列データ解析には様々な手法が提案されている．それらの手法の多くは，1つの系列もしくは予め定められた複数系列を対象として提案されている．しかしながら，現在は，様々な系列データをグループとして収集することができるようになっており，グループ化された多種多様なデータを分析の対象としてモデル化することが求められている．2つ目は，収集用に予めアクセスポイント相当のデバイスを用意する必要がある点である．安価なセンサに対して，中継機は高コストであり，設置コストの点で導入障壁が高いと言える．そこで本研究では，グループ化された系列データの解析と，設置コストを必要としない効率的な収集を目的として，以下の研究を行った．

【グループ化された系列データの解析】

グループ化された系列データの解析については，系列データを一度イベントにすることで，当該データのための汎用的な解析手法の実現を目指す．扱う対象は複数のセンサやデバイス等から取得される系列データであるが，観測された生のデータには，それぞれに特化した解析手法が必要なことが多い．しかしながら，グループ化する際に必要となるデータの観点で見ると，全ての生のデータは必要とされず，そこから抽出されたイベント系列だけで表現可能なこともある．抽象化されたイベント系列に着目し，それらをグループ化した複数系列を扱うモデルの要件を整理した．結果，(1)イベントの中身が何かとその並び，(2)イベントの継続時間，(3)イベントとイベントの間隔，(4)

イベントの重複の4つの要件を導いた．4つの要件を満たすモデルを実現するため，本稿では隠れセミマルコフモデル(HSMM)に着目し，その手法を段階的に拡張することで，4つを満たすモデルを提案する．1つ目の拡張では，系列順と系列長を考慮可能なHSMMに対して，系列間隔を扱えるようにしたDI-HSMMとIS-HSMMを提案，2つ目の拡張では，複数系列を扱う上で重要な，系列重複を扱えるようにしたOS-HSMMを提案した．各手法において，シミュレーションによりモデリング性能と認識性能を評価し，その有効性を確認した．

【系列データの収集】

系列データの収集にあたっては，想定するセンサやデバイスとクラウドサーバとの中継機であるアクセスポイント相当のデバイスを予め用意する手間とコストの削減を目指し，必要な役割を仮想的に実現することを目指す．センサやデバイスとクラウドサーバとの仲介役を担う中継機は，系列データの収集とグループ化，ある程度蓄積されたデータのサーバへのアップロードを行う．これらの役割を，対象とするエリアに存在するデバイスやモバイル端末の間で情報のやり取りと，当該端末中のストレージを借用することにより実現するアプローチを提案する．移動端末を用いる本アプローチでは，端末間の通信による端末への負荷発生，対象とするエリア内に端末が存在しなくなることによるデータの消失の可能性がある．これらを解決するために，リレータイミングの制御と対象エリアのサイズの制御を行う手法を提案する．シミュレーションにより，通信回数とデータ蓄積時間を評価し，手法の有効性を示した．

本論文で示す提案手法により，これまでに想定しなかったグループ系列やグループ系列からの新たなパタンの発見を生む可能性がある．また，グループ化された系列データの解析と収集の効率化に大きく寄与すると考えられる．

Multiple Sequential Data Modeling with Extended Hidden Semi-Markov Models and Its Data Aggregation and Management

Hiromi Narimatsu

Abstract

In recent years, with the development of devices and the development of data aggregation methods, data to be analyzed and aggregating methods have been changed. Regarding the environment of Internet of Things (IoT), sensors or devices are connected to the communication terminal as access point or mobile phone and the terminal aggregate the sensing data and upload them to the cloud server. From the viewpoint of analysis, the aggregated data are sequential data and the grouped sequence is a meaningful set of sequences because the group represents the owner's information. However, most of the researches for sequential data analysis are specialized for the target data, and not focusing on the "grouped" sequences. In addition from the viewpoint of aggregation, it needs to prepare the special terminals as an access point. The preparation of the equipment takes labor and cost. To analyze the "grouped" sequence and aggregate them without any preparation, this paper aims to realize the analysis method for grouped sequences and to realize the aggregation environment virtually. For analysis of grouped sequential data, we firstly analyze the grouped sequential data focusing on the event sequences and extract the requirements for their modeling. The requirements are (1) the order of events, (2) the duration of the event, (3) the interval between two events, and (4) the overlap of the event. To satisfy all requirements, this paper focuses on the Hidden Semi Markov Model (HSMM) as a base model because it can model the order of events and the duration of event. Then, we consider how to model these sequences with HSMM and propose its extensions. For the former consideration, we propose two models; duration and interval hidden semi-Markov model and interval state hidden-semi Markov model to satisfy both the duration of event and the interval between events simultaneously. For the latter consideration, we consider how to satisfy all requirements including the overlap of the events and propose a new modeling methodology, overlapped state hidden semi-Markov model. The performance of each method are shown compared with HSMM from the view point of the training and recognition time, the decoding performance, and the recognition performance in the simulation experiment. In the evaluation, practical application data are also used in the simulation and it shows the effectiveness.

For the data aggregation, most of conventional approaches for aggregating the grouped data are limited using pre-allocated access points or terminals. It can obtain the grouped data stably, but it needs to additional cost to allocate such terminals in order to aggregate a new group of sequences. Therefore, this paper focus on "area based information" as a target of the grouped sequences, and propose an extraordinary method to store such information using the storage of the terminals that exist in the area. It realize the temporary area based storage virtually by relaying the information with existing terminals in the area. In this approach, it is necessary to restrict the labor of terminals and also store the information as long as possible. To control optimally while the trade-off, we propose methods to control the relay timing and the size of the target storage area in ad hoc dynamically. Simulators are established as practical environment to evaluate the performance of both controlling method. The results show the effectiveness of our method compared with flooding based relay control.

As a result of above proposal and evaluation, methods for the grouped sequential data modeling and its aggregation are appeared. Finally, we summarize the research with applicable examples.

Acknowledgement

I would like to express my deepest gratitude to my supervisor, Assoc. Prof. Hiroyuki Kasai for his help and valuable discussion. He always gave me precious opportunities to discuss my research and related studies and also gave me many valuable advice at any time even on his busy schedule. His comments and suggestions were of inestimable value and they led me good direction in my life.
I am deeply grateful to Prof. Hiroyoshi Morita. He always gave me valuable advice and comments on important points. His comments always help me to improve my studies and cheered me up.
I am also deeply grateful to supervisory committee members, Assoc. Prof. Tomohiro Ogawa, Assoc. Prof. Satoshi Ohzahata, Prof. Maomi Ueno. They gave me many valuable comments for my research and all of them help improving my research. My special thanks to Prof. Yasuhiro Minami, and Assistant Prof. Akiko Manada. They also gave me precious suggestions and cheered me up.
I am also express my special thanks to Assistant Prof. Ryoichi Shinkuma at Kyoto University. The discussion with him and his Lab. members inspired me and gave valuable comments for my study.
I am grateful to Prof. Hiroshi Watanabe and Prof. Wataru Kameyama at Waseda University. They gave me valuable feedback for my research. The discussion with their Lab. members also inspired me to conduct my study.
I am also grateful to the lab assistant Ms. Makiko Katagiri also support me to my student life. My Lab. members also cheered me up for my research and also gave me questions and suggestions for my research presentation. My friends also cheered me up and gave me good stimuli for my motivation to complete this study. I would also like to express my gratitude to all surrounding me including my Lab. members and friends.
I wish to thank the members of my workplace at Nippon Telegraph and Telephone Corporation. Especially I would like to thank my bosses, Mr. Hiroshi Jinzenji, Mr. Kazuo Kiramura and Dr. Hiroshi Sawada for their warm care.
I would like to thank all reviewers reviewing our paper to the conference or journals. They also gave me special and valuable comments to improve my study.
One of the part in Section 5 is supported in part by the National Institute of information and Communications Technology (NICT), Japan, under early-concept Grants for Exploratory Research on New-generation network.
This paper was completed by not only me but also all of the persons who supported me. I would like to special thank you for all.
Finally, I would like to give special thanks to my family for their moral support and warm encouragement. Without the support from my family, I couldn't have proceed my studies to the completion.
At the end of the acknowledgement, I would like to appreciate again for all professors and staffs to cooperate my submission and preparation of this preliminary examination. Thank you for reading this paper on your busy schedule, and thank you for giving me the opportunity to give this research presentation for preliminary examination.
At the end of the acknowledgement, I would like to represents my determination for my future. All the experiments and comments from many supporters in my study are precious gifts for my life. I would like to cherish the precious gifts and I would like to contribute to provide new values, new technologies, new insight in the future. I will make even greater efforts for the contribution.

December 2017
Hiromi Narimatsu

Contents

1 Introduction **1**
- 1.1 Background . . . 1
- 1.2 Studies for sequential data analysis . . . 2
  - 1.2.1 History of sequential data studies . . . 2
  - 1.2.2 Target of sequential data analysis . . . 5
- 1.3 Studies for sequential data aggregation . . . 6
  - 1.3.1 History of data aggregation in the physical field . . . 6
  - 1.3.2 Target of data aggregation . . . 7
- 1.4 Outline . . . 8

2 Studies for sequential data analysis **9**
- 2.1 Motivation . . . 9
- 2.2 Related work . . . 10
- 2.3 Requirement verification and basic model . . . 12
  - 2.3.1 Requirement verification for conventional models . . . 12
  - 2.3.2 Hidden semi-Markov model (HSMM) . . . 13
  - 2.3.3 Notations . . . 14
  - 2.3.4 Model training (inference) in HSMM . . . 14
  - 2.3.5 Recognition using HSMM . . . 19

3 Duration and interval modeling with HSMM **21**
- 3.1 State interval modeling in HSMM . . . 21
  - 3.1.1 Two approaches for state interval modeling . . . 21
  - 3.1.2 Problems of state interval modeling . . . 22
- 3.2 Interval Length Probability HSMM (ILP-HSMM) . . . 24
  - 3.2.1 Model training (inference) in ILP-HSMM . . . 24
  - 3.2.2 Recognition using ILP-HSMM . . . 26
- 3.3 Interval State Hidden Semi-Markov Model (IS-HSMM) . . . 29
  - 3.3.1 Model training (inference) in IS-HSMM . . . 29
  - 3.3.2 Recognition using IS-HSMM . . . 32
- 3.4 Evaluations . . . 34
  - 3.4.1 Experimental data . . . 34
  - 3.4.2 Execution time evaluation . . . 34
  - 3.4.3 Recognition performance evaluation . . . 35
  - 3.4.4 Reproducibility performance evaluations between IS-HSMM and ILP-HSMM . . . 37
- 3.5 Summary . . . 39

4 Overlapped state hidden semi-Markov models **43**
4.1 HSMM with multiple sequence input . . . 43
4.2 Overlapped State HSMM (OS-HSMM) . . . 44
4.2.1 Model training (inference) in OS-HSMM . . . 44
4.2.2 Decoding and recognition using OS-HSMM . . . 47
4.3 Evaluation . . . 48
4.4 Summary . . . 51

5 Grouped sequential data aggregation **57**
5.1 Studies for data aggregation and management . . . 57
5.1.1 Motivation . . . 57
5.1.2 Related work . . . 60
5.1.3 Difference of the related research . . . 61
5.2 Concept of area-based collaborative mobile storage . . . 62
5.2.1 Proposal basis . . . 62
5.2.2 WLAN AP-based storage area management . . . 63
5.2.3 Platform architecture overview . . . 64
5.2.4 System behavior . . . 66
5.3 Details of relay control algorithm . . . 67
5.3.1 Problems and requirements for relaying store data . . . 67
5.3.2 Relay timing and terminal determination algorithm . . . 67
5.3.3 Relay area control algorithm . . . 69
5.4 Simulation evaluation . . . 71
5.4.1 Simulation method . . . 71
5.4.2 Experiment results . . . 72
5.5 Other consideration and application example . . . 90
5.5.1 Other consideration . . . 90
5.5.2 Application examples . . . 92
5.6 Summary . . . 94

6 Conclusion **95**
6.1 Summary . . . 95
6.2 Future work . . . 96

Publications **103**

Chapter 1

Introduction

1.1 Background

In recent years, with the development of devices and the development of data aggregation methods, the target data to be analyzed and the aggregation methods have been changed. For example, some of the agriculture services enable remote monitoring and automatic adjustment of water quantity control using temperature sensors and humidity sensors on the farmland [1]. Agricultural monitoring using sensors has been spread conventionally, but in recent years the number of sensing data has been increasing according to the development of sensors. For example, not only temperature but also humidity, solar radiation, soil moisture, leaf wetness, ultraviolet light, CO2 amount, pest counter, etc., are used for monitoring. The system for such monitoring services consists of sensors, application servers on the cloud, and the relay nodes that connect with sensors and the cloud server via network communication. The sensing data are aggregated to the server via the relay nodes and analyzed for its management services. Healthcare monitoring services also use sensors or wearable devices to support user's healthcare. For example, smart watches and fitness devices monitor user's behavior and connect with user's smart phone to manage the observed data [2]. Although such devices do not have the function communicate with the server on the cloud directly, the smartphone connects with the devices and the server via the application in itself.

The above is a partial example, but there are two things in common between these services. The first one is that data are sent to the server on the cloud via one or more devices that play like an access point. The second one is the aggregated data from various sensors to an access point consists a group that is meaningful. Example of devices consists of such services, i.e., the agricultural monitoring service and the healthcare services with wearable devices, are shown in Figure 1.1 and Figure 1.2. In the agricultural services, the relay nodes communicate with sensors via short-range wireless communication, and the sensing data is relayed to the cloud server via the relay nodes. In the healthcare monitoring services, the wearable devices communicate with smartphone via Bluetooth communication and the sensing data relayed to the cloud server via the smartphone. The relay node or the smartphone plays a role of an access point that communicates with the cloud server. Besides, the aggregated data are managed along with the device's ID as an access point and they are managed and analyzed in combination with the data related with other data with the IDs in a predefined group.

Both analysis and aggregation technologies have the following problems. One is for the analysis that there is no suitable method to treat a group of multiple sequences as the target. The data aggregated to the cloud server every moment are characterized by "time series." Most of the analysis methods are specialized for each target sequence or multiple sequences. However, it is possible to

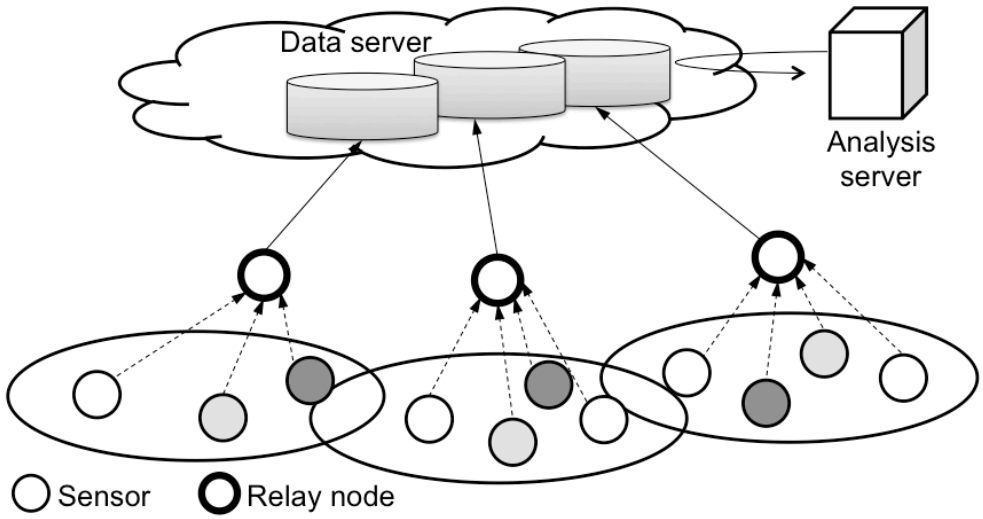


Figure 1.1: Example device allocation for location status monitoring.

aggregate multiple sequences as a target group currently, and it is required to model the group of multiple sequences as a target. The other is for the aggregation that it needs to prepare or allocate the device that plays a role as an access point. The cost for preparing relay node or smartphones is more expensive than other sensors or devices. Therefore, the cost for preparation is an obstacle for starting such services using sensors or devices.

To tackle the problems mentioned above, this study proposes a general-purpose time series data analysis method and proposes a grouped data aggregating method that does not require preparation of access points in advance. For each subject, researches that have been conducted in the individual field as an individual field for a long time. Explain each research and its adaptability.

1.2 Studies for sequential data analysis

Sequential data studies are affected by the development of various technologies such as sensing, recording technologies of devices. Section 1.2.1 describes the history of sequential data studies and Section 1.2.2 describes the goal of our sequential data analysis and explains our proposed method briefly.

1.2.1 History of sequential data studies

Sequential data analysis has been studied for a long time for the purpose of analyzing the stock price fluctuation, sunshine hours estimation, weather estimation and so on. The studies began around 1960 and these have three origins. The first one is Autoregressive (AR) model that predicts the next observation value from the past sequential data in a single continuous sequential data by matching partial patterns [3]. The second one is the Dynamic Programming (DP) Algorithm [4] that calculates the degree of matching between the target sequential data and the reference sequential data. The third one is Kalman filter [5] that supposes that the observed values changed depends on the potential internal state, and estimate the state sequence from the observation sequence even though the data contains missing values or noise. The studies can be classified into three categories from the viewpoint of the modeling method. The first one is a model for continuous sequential data that does not allow

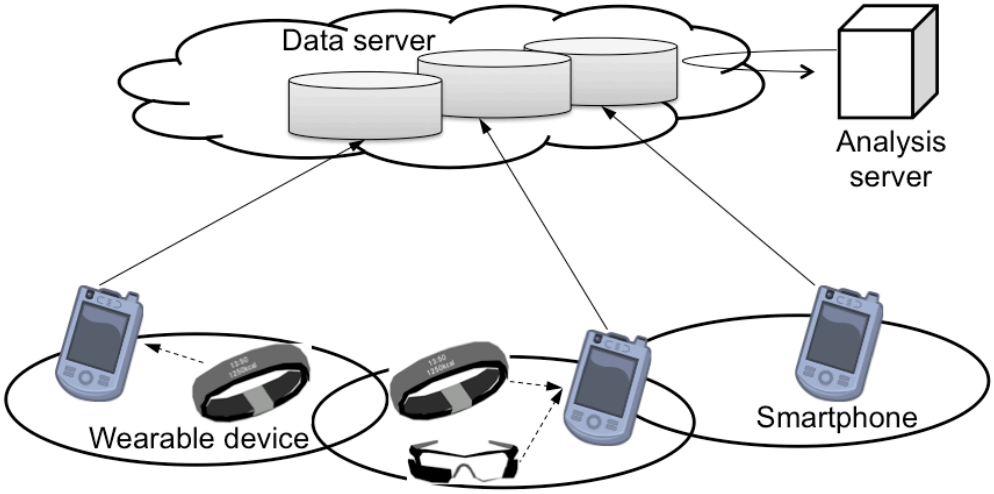


Figure 1.2: Example device allocation for healthcare monitoring with wearable devices.

missing in the data such as AR model. The second one is matching method of sequential data such as DP matching. The third one is a state space model that allows noise or missing in the data such as Kalman filter. For each classification, since 1960 onwards, many methods of various kinds are studied and proposed according to the kinds of collected data.

For the first category, Autoregressive (AR) model is proposed by Akaike in 1969 as origin [3] and various extension model has been proposed followed by AR model. The AR model expresses the possibility that values at a past time point on the same time series influence the current value, in brief, there is a correlation, and it assumes that data can be expressed as a linear combination. It is used for the economic change data analysis, engineering data analysis, and it is the basic model to control the own system automatically by estimating the own state from the observation results. There is also a simple Moving Average (MA) model that models the average of the differences from the past, and the model is used often combined with the AR model named the Autoregressive Moving Average (ARMA) model [6]. Furthermore, these models are extended to treat nonlinear data or other target data. The models that are applied AR model to non linear data have been proposed [7, 8] and another model that is combined AR model with neural network model has also been proposed [9]. In the 1990s, Singular Spectrum Analysis (SSA) has been also proposed to analyze the structural change of data itself without assuming a specific model [10]. SSA is proposed for the purpose of removing noise from observation signals of sounds, and it is possible to separate and extract major fluctuating components and unsteady fluctuating components. In recent years, since it is possible to constantly acquire browsing history and purchase records on the Web and centrally manage large amounts of data, the expectation for rich browsing, reference and data analysis has increasing. From this background, Multiple-SSA (MSSA) has been proposed to deal with large amount of data from massive users simultaneously. MSSA is used to extract trends from market purchasing data or extract trends from social data. The method is also combined with Tensor to analyze hobby and tastes, and it is also applied to estimate and score the sleeping status [11]. The characteristics of the models and analyzing methods are measuring the difference between the past value and the current value, and targeting the continuous single sequential data.

For the second category, various extended methods have been proposed since Dynamic Programming (DP) Algorithm [4] was proposed in 1957. It is slightly different from the range of modeling because it deals with the matching measurement. The DP algorithm deals with a problem of dividing

the sequential data into partial sequence, calculating each matching score of each partial sequence, and calculating the whole matching score by using the partial results. For example, with respect to matching the speech recognition results and the target data, the confidential score of word recognition results is estimated by all mora recognition results. DP matching method had been often used for speech recognition at the beginning. The extended DP algorithm, named A Dynamic Time Warping (DTW) method was proposed to allow the time difference in speech time in every human and deal with the time constraint [12, 13, 14]. In speech recognition, Hidden Markov Model (HMM), that was later proposed, became mainstream later. Instead of that, DTW in recent years has been applied to human gesture recognition from video images and behavior estimation [15, 16]. These methods deal with discrete data in many cases, but the extended algorithms for dealing with continuous sequence have been proposed. DP matching and DTW are characterized not as a statistical modeling method but as a method for evaluating matching score between sequences.

Third category, state space model, includes the Kalman filter proposed in 1960's [5] and HMM. Kalman filter is proposed by Rudolf Emil Kalman and the model estimates the potential state from the observation sequence values even if it includes noise or missing values. It modifies and updates the inner state after obtaining the current observation data (posterior data), then it can reduce the estimate error. It is used for rocket's posture control. Although AR model assumes linear model, some extended algorithms have been proposed for dealing non-linear model [17]. The extended models have been used for robot's self position estimation and mobility tracking using the Doppler shift of the radio frequency [18]. In recent years, they have also been applied for measuring the influence of radio interferometry [19]. Although the Kalman filter is not effective for non-Gaussian data, Monte Carlo filters were proposed for non-Gaussian data according the types of datum acquired in the real fields [20]. The parameter estimation algorithm for HMM, *Baum Welch algorithm*, is also proposed in 1960's [21]. HMM assumes to have potential states towards observation sequences, and it estimates the state sequence for each observation sequence. Therefore, it is profitable for a patterned data such as weather data or average date temperature in a year. Especially, HMM has been well known and studied in the speech recognition research. It is useful for Part-of-Speech (POS) tagging, since the order of the words or character units in a word is restricted by the grammatical feature. In 1989, HMM were applied for DNA segmentation for the first [22], then it has been widely applied for DNA phylogenetic tree estimation, biological data analysis, and other statistical sequential data analysis [23]. Afterwards, the model has also been used for detecting human activities from video images in sports [24, 25]. In recent years, it has been used in the field of modeling health monitoring, on-line learning for tracking objects from images of surveillance cameras such as monitoring and recording all the time, detection of malware from network usage logs acquired from time to time every moment according the types of target data [26, 27]. Then, further integration models have been proposed for respective applications. A hybrid method of Recurrent Neural Network (RNN) and HMM [28] and a hybrid method of SVM and HMM have been proposed and used for improving accuracy of speech recognition of Arabic (Arab) [29]. Also, various HMM extension methods have been proposed according to the application. There are many extended HMM, for instance, IOHMM that considers the mapping of input and output, EDM and HSMM that deal the duration time staying in a state toward high precision of speech recognition and handwritten character recognition, Hierarchical HMM that handling state sequence with hierarchical relations, etc. The characteristics of these models are handling discrete data, and they are probabilistic models that can take into account the state space with high degrees of freedom.

As described above, the sequential data analysis has been widely applied for various kinds of application data according to the development of recording, aggregating, and storing in devices and servers in cloud network, and various extended methods have been proposed.

1.2.2 Target of sequential data analysis

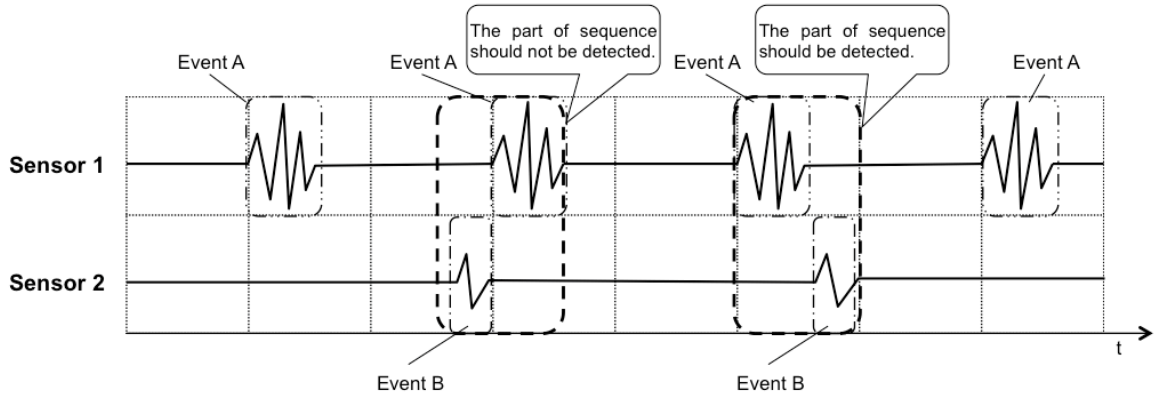


Figure 1.3: Target data.

The goal of our sequential data analysis is that the grouped sequences can be modeled. It is difficult to gather the row sequential data as a group directly. Each sequential data needs its modeling method specialized for the characteristics. Since each data has different characteristics, it is required to be analyzed individually in order to understand each waveform of raw data as an event. However, not all observed values are necessarily required when grouping sequences. One example is a life log data. We assume the behavior pattern of person A as "Person A always get a newspaper from a mail box just after waking up in the morning, then brush his teeth, and read the newspaper." In case that three sensors, the sensor S1 attached to the mailbox, the sensor S2 attached to the toothbrush, and the wearable sensor S3 like smart watch attached to the user's wrist, the information that needs to be held by each sensor is "the toothbrush was used (E1)," "the mailbox was opened (E2)," "action turning over the newspaper was observed (E3)." That is, it is possible to express a grouped sequence by the patterns only using the order of events E1, E2, E3 and the duration of the event and the interval of events. Therefore, we focus on the pattern of event occurrence and aim for modeling the grouped abstracted sequences. Since the target of modeling is new, we first analyze the grouped sequences and extract the requirements for modeling. Figure 1.3 shows an example of the target data grouping two sequences from sensors. As you can see from the figure, it is figure out that it is possible to represent the grouped sequences using the order of events (R1), the duration of the event (R2), the interval of events (R3) and the overlap of events (R4) by abstracting the observed data as an event. To realize the model satisfying the above four requirements, this research focuses on hidden semi-Markov model (HSMM) that can mode the events order and the duration of an event and proposes new extended models.

HSMM can model the order of events (R1) and the duration of an event (R2) by representing each event as a state. To simple explanation, these four requirements are *states order* (R1), *state duration* (R2), *states interval* (R3), and *states overlap* (R4) using state instead of events in other words. The extended models to satisfy these requirements step by step, first the models satisfying the state duration and the states interval simultaneously are proposed, and then the further extended model is proposed to satisfy the state overlap.

To model the state interval, there are two approaches: first approach is representing the state interval as an *interval state* that emits a *no observation* symbol, and the second approach is representing the time length of the state interval as an interval length probability directly by introducing a new parameter. For the first approach, there is a problem that bias of the transition probability

occurs when there are many states intervals. To solve the problem, we propose a new method to model the transitions including the interval state by using second-order Markov chain partially in order not to be affected by the bias of transitions from / to the interval state. For the second approach, there is a problem of how to use the new parameters. We propose a model that expresses the interval length probability stochastically. Each method is implemented and its effectiveness is shown.

To model the state overlap, it is necessary to consider how to handle multiple sequences using a model that allows only one sequence input. It is also possible to assume that modeling each sequence individually and combining these models next. However, this approach has following problem. If there is only one observable event per sequence, it is impossible to model the sequence accurately. This approach cannot model the relationship of events observed in multiple sequences. To solve this problem, we propose an approach to combine the grouped sequences in advance. However, combining the multiple sequences is not simple approach because events may occur in multiple sequences at the same time. It can be considered that two events are combined as a new event, but it is not realistic because the number of observable symbols must be determined *a priori* in HMMs and the number of all possible combined event patterns are huge. Therefore, we propose a method to treat overlapping parts as a sequence by shifting the sequence so that they do not overlap. Although shifting seems to be a peculiar method considering the possibility of decoding, we devise that giving overlapping labels to overlapping parts and modeling the shifted time length to model the state overlap that can be decoded. We implemented this method and showed its effectiveness.
Thus, we realize a model that satisfies the above four requirements necessary for modeling the grouped sequences.

1.3 Studies for sequential data aggregation

Data aggregation studies are also affected by the development of various technologies of devices and communication. Section 1.3.1 describes the history of data aggregation studies in a physical field and Section 1.3.2 describes the goal of our data aggregation and explains our proposed method briefly.

1.3.1 History of data aggregation in the physical field

The study for the sensor network has been studied since 1970's. There are two purposes for the research; one is for clock synchronization between unmanaged systems (terminals), the other is for data transmission and reception method in order to acquire event information in the distributed system [30]. At that time, data transmission and reception between remote locations via the Internet because the Internet became widespread gradually, but since the server area was also small, research for data management by the distributed system has been studied [31]. They have been studied under the name *concurrency control*. In the 1980's, besides time synchronization, research on allocation and update timing has been studied in order to allocate / update the replicas considering the redundancy and the guarantees of the data safety in distributed systems [32]. After that, research on hierarchical data collection and routing for each point in a large-scale network was started, and research on sensor networks began with the advent of sensors in the 1990's [33, 34, 35]. Because of this background, the sensor network assumes the time synchronization in the network, and the research on the time synchronization method [36] and the routing method has been actively studied [37]. From 2000 onwards, it is sometimes called a wireless sensor network, emphasizing the word "wireless" in contrast to distributed networks studied before. The sensor network consists of sensors, routers, and gateways, and it is necessary to arrange the sensors so that it can always communicates with one of the routers

in the network. Also, when one relay fails, another route is assigned, so research on failure detection and routing has also been studied [38]. However, it is difficult from the viewpoint of cost to arrange redundancy for many routers for the case of routers' malfunction. Therefore, in recent years, a method of collecting information of fixedly arranged sensors by a mobile repeater and a routing method have been studied [39]. These have been studied mainly on IEEE 802.15.4 based communication system that is optimum for sensor communication.

On the other hand, research for ad hoc communication has been studied since the 1990's when a new protocol [40] for ad hoc communication in Wi-Fi was proposed. In recent years, it is called ad hoc mode in Wi-Fi and the method is that terminals in the same Local Area Network can mutually send and receive data. Unlike the sensor network described above, it assumes that each node is mobile, and temporarily configures the network [41]. Recently, in addition to IEEE802.11, Bluetooth and HyperLAN can also be used for ad hoc communication, so it has high versatility and has been studied as a temporary structure of the communication network at the time of disaster. On the other hand, if the mobile terminal itself is a malicious terminal, there is a possibility of intercepting information and posing a threat to the network. From this fact, a method to evaluate the reliability of the terminal when constructing the network has also been studied recently [42, 43]. As mentioned above, while it is possible to guarantee the reliability of the network and terminals in the sensor network, it is difficult to participate in the mobile terminal, also temporary, there is a disadvantage, data transmission and reception in ad hoc communication has the merit that a general-purpose terminal can join because it configures the network temporarily on the spot temporarily data to be applied to reliability.

1.3.2 Target of data aggregation

The goal of the data aggregation is aggregating a grouped data without preparing the dedicated access point in advance. To realize that, we consider mobile terminals play a role of an access point virtually. The role of an access point in the target situation is aggregating the sequential data from the sensors allocated in the area, and sending the necessary data extracted from the aggregated data to the cloud server via network after accumulated the data on a certain extent. In order to play the role using mobile terminals, it is necessary to aggregate and store the data in the target area under the circumstance that the terminal moves. To solve the problem, we propose an approach aggregating the data in the existing mobile terminals in the target area and relaying the data to store them. The concept of this approach is similar to the ad hoc sensor network, but the purpose is different. The purpose of ad hoc sensor network is relaying a target data from the source to the destination. It is not necessary to store the data for a while. The purpose of our target is relaying the data to store it in the target area. There is no destination terminal in the situation. Storing the data in the target area as a virtual storage via mobile terminals is a new challenge in the research field. Therefore, we propose a new method how to relay the data and how to store the data effectively in the target area. This approach has two problems. One is the situation that there is no terminal in the target area. If there is no terminal, the data is vanished. Therefore, it is necessary to store the target data even if there is no terminals temporary. The other is that the method loads to the terminals when relaying the data. Relaying via wireless communication drain the battery of the terminals. To solve these problems, a relay controlling method and a relay area controlling method are proposed. The relay controlling method determines whether the terminal should relay the data to the passed terminal according to the situation. The relay area controlling method determines the target area not to occur the situation that there is no terminal in the target area as far as possible. We implemented this method and evaluate the effectiveness in the simulation.

1.4 Outline

Finally, outline of this paper is described. In Chapter 1, we described the background of deciding the purposed of our thesis as a realization of a model that can deals with multiple sequences and the methods to its data aggregation and management. In Chapter 2, we describe the related work for sequential data analysis and extract the requirements for models that can deal with multiple sequential data. In Chapter 3, we first consider the model to deals with duration and interval in a sequence simultaneously. We proposed two models; interval length probability HSMM (ILP-HSMM) and interval state HSMM (IS-HSMM). Then, in Chapter 4, we further expanded one of the model proposed in Chapter 3 to deal with the overlap of events. The proposed model named overlapped state HSMM (OS-HSMM) makes it possible to deal with multiple sequences. The evaluation results are also described and they show the effectiveness of the proposed model. In Chapter 5, we describe the requirements for aggregating and managing the grouped sequences, and show the proposed concept. Then, we propose a new method to aggregate area based information without pre-allocated terminals. We propose two controlling methods; relay timing control method and relay area control method, in order to realize the concept of collaborative mobile storage. The effectiveness of our methods are also shown by simulation experiment in this section. Finally in Chapter 6 we summarize this paper.

Chapter 2

Studies for sequential data analysis

2.1 Motivation

This section presents an analysis of sequential data modeling and derives the model requirements for sequential data analysis. Then, the satisfactions of the extended models of HMM for the model requirements are examined. This section presents discussions of the requirements for model description using time-series data: representative data of sequential data. For this purpose, we assume a situation in which multiple different sequences are generated independently from five sensors as shown in Figure 2.1. Here, an observed event of which value exceeds a predefined threshold is recognized as a 'state' represented in a block. The continuous period of each event is represented by the block length. Because events are not successively observed, a *no-observation period* exists between two successive states in certain periods. The length of such a no observation period is represented as the distance between two blocks. In this example, we also assume that a set of four black blocks, $\{S_1, S_2, S_3, S_4\}$, expresses an extracted multiple states that forms one particular group.

Now we extract the requirements for model description. First, addressing this formation of four blocks, it is readily apparent that these states are observed in a prescribed order. Therefore, it is apparent that the order of multiple states should be described in a model (**R1**). Second, multiple states are visible in a partially overlapped manner, as shown by S_1 and S_2. In other words, multiple states can occur simultaneously at a certain period. Therefore, the model must support the representation capability to describe multiple states occurring at the same time (**R2**). Third, because the time lengths of respective states mutually differ, the state duration must be expressed in a model (**R3**). Finally, for the case in which each state occurs intermittently, a vacant period between one state and another state that is not involved in the group of sequence might exist between two states. Furthermore, the length of this vacant period shall be variable. Therefore, the state interval between two states in a model must be described (**R4**). In summary, the sequential data model is required to describe these requirements. This report defines these respective requirements as follows.

(i) **R1:** State order

(ii) **R2:** Staying multiple states in a certain period

(iii) **R3:** State duration

(iv) **R4:** State interval

Among these items, **R2** differs from other items because **R1**, **R3**, and **R4** are required even for a single sequence, whereas **R2** is the requirement for multiple sequences. Therefore, this study specif-

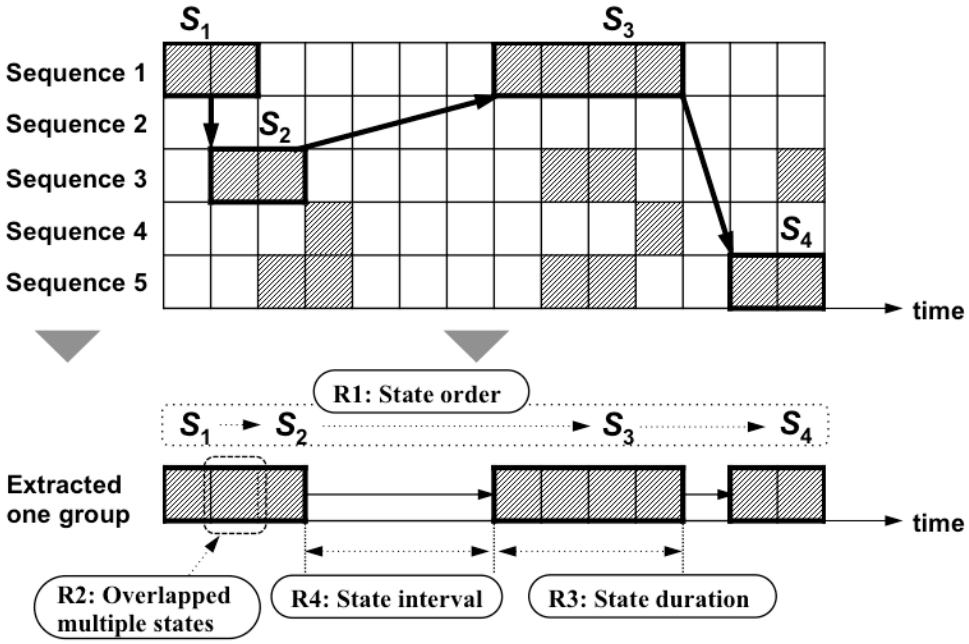


Figure 2.1: Event generative model and sequential data model requirements.

ically examines requirements **R1**, **R3**, and **R4**. The examination of **R2** shall be left for advanced studies to be undertaken as future work.

2.2 Related work

This section presents related work that has been reported in the field of sequential data analysis. For sequential pattern matching and sequential pattern detection, the Dynamic Programming (DP) algorithm [12] provides an optimized search algorithm that calculates the cost of a path in a grid and which thereby finds the least costly path. Actually, DP was first used for acoustic speech recognition. For sequential pattern classification, Support Vector Machine (SVM) [44, 45] is a classifier that converts an n-class problem into multiple two-class problems. SVM has demonstrated its superior performance in a diverse applications such as face and object recognition from a picture. Regarding the Regression Model (RM) [46], the logistic regression model [47] is a representative model that is powerful binary classification model when the model parameters are mutually independent. The hidden Markov model (HMM), originally proposed in [21, 48], is a statistical tool used for modeling generative sequences. HMM has been used frequently together with the Viterbi algorithm to estimate the likelihood of generating observation sequences. Whereas HMM is used widely for many applications such as speech recognition, handwriting recognition, and activity recognition, many extended HMMs have also been proposed to enhance the expressive capabilities of the baseline HMM model and to support various specialized application data. Consequently, addressing HMM as a powerful and robust model for treating sequential data using its transition probability in a statistical manner, we particularly examine HMM in the present paper.

With regard to the extensions of HMM, Xue *et al.* proposed transition-emitting HMMs (TE-HMMs) and state-emitting HMMs (SE-HMMs) to treat the discontinuous symbol [49], of which application is an off-line handwriting word recognition. The observation data include discontinuous

and continuous symbols between characters when writing in cursive letters. They specifically examined such discontinuous features and continuous features, and extended HMM to treat both. Bengio *et al.* specifically examined mapping of input sequences to the output sequences [50]. The proposed model supports a recurrent networks processing style and describes an extended architecture under the supervised learning paradigm. Salzenstein *et al.* dealt with a statistical model based on Fuzzy Markov random chains for image segmentations in the context of stationary and non-stationary data [51]. They specifically examined the observation in a non-stationary context, and proposed a model and a method to estimate model parameters. Ferguson proposed a variable duration models of HMM for speech recognition. Today, the model is familiar as the extended model of HMM as explicit-duration hidden Markov model or hidden semi-Markov model [52, 53, 54, 55]. They proposed a new forward–backward algorithm to estimate model parameters.

Addressing the difference of duration in each state, hidden semi-Markov model (HSMM) is proposed to treat the duration and multiple observations produced in a single state [56, 57]. The salient difference between HMM and HSMM is whether it can treat the duration of states in HMM. The technique of EM algorithms for modeling the duration of states was proposed by Ferguson [58]. He proposed the algorithm for speech recognition, but the model is further applied for time-series data for word recognition and rainfall data [59, 60, 61, 62]. Then, Bulla proposed an estimation procedure to the right-censored HSMM for modeling financial time-series data using conditional Gaussian distributions for the HSMM parameters [63, 64]. For diagnosis and prognosis using multi-sensor equipment, Dong *et al.* prioritized the weights for each sensor to treat multiple sensor results, and showed that the proposed model of HSMM gave higher performance than the original HSMM [65]. Recently, Dasu analyzed HSMM and described how to implement HSMM for a practical application in detail [66]. Baratchi *et al.* and Yu *et al.* proposed expanded hidden semi-Markov models for mobility data. [67, 68] These models can treat the sequential data which include missing data.

Researches for treating sequential data have been studied for decades. Dynamic Programming (DP) algorithms are studied for sequential pattern matching and detection [12]. The algorithms are applied for human behavior recognition, image classification, and biological data pattern detection. Esmaeili *et al.* [69] categorized sequential patterns in three types; periodic patterns, statistically significant patterns, and approximate patterns from the view point of theoretical investigation. All of the patterns are important for sequential data modeling. Shimodaira *et al.* applied Support Vector Machine (SVM) to sequential-pattern recognition [70]. They propose a new time-alignment operation in kernel function, and it can be applied for speech recognition without any modification in classification algorithm. Baum *et al.* studies statistical inference and estimation for Hidden Markov Models (HMMs). [21, 48] The models are widely used in speech recognition, handwriting recognition, and human activity recognition.

These algorithms are applied for treating the time length of patterns and multiple sequence input respectively. For treating time length, DP algorithms are extended to treat dynamic time warping [71]. These algorithms are also applied for multiple sequence alignment in the biological field [72]. SVM are also treating time delay in time series prediction [73]. Hidden semi-Markov models (HSMMs) are proposed to extend HMMs to treat a time length, i.e., "duration" in staying the same states for improving the recognition performance for speech recognition and handwriting recognition [56, 57]. The models are further studied to treat multiple observation input [74].

Considering these methods from the view point of handling both time length in patterns and multiple sequence input, DP algorithms and HSMMs are suitable as a basic method for our approach. Furthermore, since there exist many extended models of HSMMs, we examine HSMM on basis.

Some extended models for treating multiple sequence input for HSMM. Natarajan *et al.* studied a coupled HSMM for activity recognition where the relatives between an state from a sequence and

Table 2.1: Requirement satisfactions in HMM, HMM variants, and our proposals.

Model	Requirements	
	Time length in a state (**R3**)	Time Interval between states (**R4**)
HMM (baseline) [78]		
HMM-selftrans [49]	✓	
FO-HMM [51]		
IO-HMM [50]		
EDM [52, 53] and HSMM [56, 57]	✓	
IS-HSMM and **ILP-HSMM** (proposal)	✓	✓

another state from difference sequence exist. [75] The symbols in a target data are observed all the time, so it does not assume the time space in the sequence. They also studied hierarchical HSMMs to treat the abstract states in a model [76]. They propose two-layer structured HSMM, and each state has a HSMM in the model. They focus on the abstracted event sequence. It treats nest structured models. However, it also does not assume the multiple sequences with time space. Recent studies for expanding HSMMs focus on the time space [77]. They treat not only duration but interval i.e., time length between states. It can treat the time interval in a sequence, but its target is not multiple sequence inputs.

Therefore, to treat multiple sequence with time length in the sequence, we deal with HSMM as a basic model of our approach. Next section introduces HSMM in detail, and describes the problems to treat multiple observation sequences.

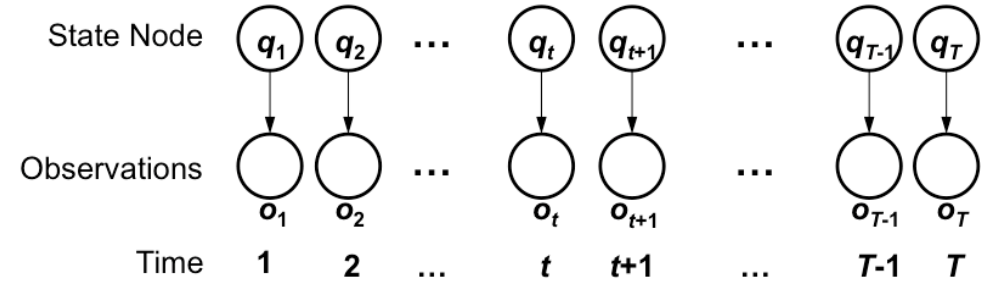
2.3 Requirement verification and basic model

2.3.1 Requirement verification for conventional models

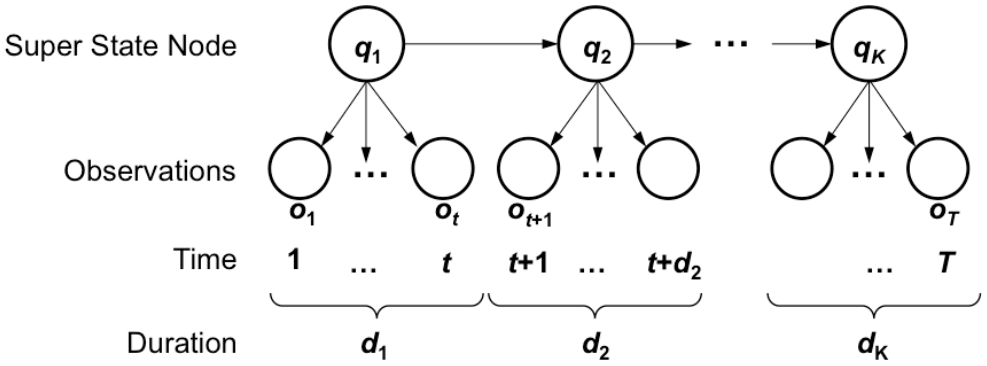
This section presents investigation of whether HMM and the extended variants of HMM satisfy those requirements. Table 2.1 presents a comparison among the existing HMM models from the viewpoints of the model requirements described above. Because the baseline HMM model describes the order of the states (**R1**), all the extended HMM models inherit this capability. FO-HMM is specialized for treating the ambiguity of observation symbols. It does not contribute to our model requirement. IO-HMM is a hybrid model of generative and discriminative models to treat the estimation probability commonly used for input sequence and observations. Therefore, it does not satisfy the remaining requirements. HSMM models the time length to remain in a single state [56]. Its variants including HMM-selftrans and EDM [52, 53, 54, 55] satisfy the same requirements: state order (**R1**) and state duration (**R3**).

As a result of investigation of the requirement satisfaction, it is apparent that no existing HMM model accommodates both the state duration and state interval together. Nevertheless, we conclude that HSMM is the best baseline model to be extended towards our new target model because only HSMM handles state duration.

Moreover, some expanded models of HSMM have been proposed. Baratchi *et al.* and Yu *et al.* proposed expanded models of HSMM that can treat missing data. Their proposal can model the sequential data even if they include missing intervals [67, 68]. These studies are motivated to complement the missing data so that the 'interval of missing' might have variable status in all sequences. It is useful for modeling even if it has missing data and completing the missing data. However, in the situation we lead from the sequential data analysis described in this section, the



(a) Model structure of HMM. The state node of HMM emits an observation symbol.



(b) Model structure of HSMM. The super state node of HSMM emits observation sequence in a certain duration.

Figure 2.2: Model structure comparison between HMM and HSMM.

interval is not 'missing'. The status of the interval is only the interval which includes other status that is unrelated to the sequence. Therefore the target for modeling differs. It is necessary to model the *interval* which is not missing. Therefore, the next section provides a detailed explanation of HSMM.

2.3.2 Hidden semi-Markov model (HSMM)

HMM has been studied as a powerful model for speech recognition. The model parameters of HMM consist of the initial state probability, the transition probability between states and the emission probability of observation elements from each state. The model training phase calculates the optimum values of the model parameters. The recognition phase calculates the probabilities that generates an observed sequence for each model, and then selects the highest probability model as a recognition result.

The distinguishing feature of HMM is to model the transition probability of every pair of two states. However, the time length to stay in each state cannot be modeled by HMM, which is fundamentally necessary for modeling in some useful applications such as online handwriting recognition. HSMM, which has been proposed to support this time length, has long been studied for some specific applications such as speech recognition and online handwriting recognition. This section, after providing basic notation, presents details of the algorithms of the model training and recognition in HSMM.

2.3.3 Notations

The structures of HMM and HSMM are shown in Figure 2.2. A set of output symbols is expressed as $Y = \{y_1, y_2, \cdots, y_N\}$, where N is the number of symbols. The observation sequence is $o_1 o_2 \cdots o_T$ and $o_t \in Y$ is the observation at time t. Likewise for hidden state, a set of hidden states is S, the number of states is M, and the hidden sequence is $q_1 q_2 \cdots q_T$ and the hidden state at time t is $q_t \in S$. For simply writing, i-th state and j-th state in S is represented as $i, j \in S$ respectively. Each state q_t emits an observation symbol o_t. The parameters to be estimated are transition probability from state i to state j represented as $a_{i,j}$ and it is defined as,

$$a_{i,j} \quad := \quad P(S_{t+1} = j | S_t = i). \tag{2.1}$$

The emission probability of y_n from state j represented as $b_j(y_n)$, and it is defined as

$$b_j(\boldsymbol{o}_{t+1}) \quad := \quad P(\boldsymbol{o}_{t+1} | S_{t+1} = j). \tag{2.2}$$

Note that $\boldsymbol{o}_{t+1}$ is represented in $b_j(\boldsymbol{o}_{t+1})$ using t, but it stands for the observed symbol $y_n \in Y$ at time $t+1$ for all t. The initial state probability of i is represented as π_i which is the same as a_i. Each parameter matrix is represented as $\mathbf{A} \in \mathbb{R}^{M \times M}, \mathbf{B} \in \mathbb{R}^{M \times N}, \boldsymbol{\pi} \in \mathbb{R}^M$ respectively and the set of parameters is $\Lambda = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$. On the other hand, the hidden sequence of HSMM is $q_1 q_2 \cdots q_K$ and K is the number of hidden states in the sequence. Each k-th hidden state q_k can emit not only an observation symbol but also multiple observation symbols. The significant difference between HMM and HSMM is the characteristic of q_k stays for duration d. Therefore, all of the parameters are represented using the duration d.

To deal with the duration of each state, the notations are newly defined in HSMM. The observation sequence from time $t = t_1$ to $t = t_2$ is denoted as $\boldsymbol{o}_{t_1:t_2} = o_{t_1}, ..., o_{t_2}$. The hidden state sequence from time $t = t_1$ to $t = t_2$ is expressed as $S_{t_1:t_2} = S_{t_1}, ..., S_{t_2}$, where S_t represents a state at time t. The k-th hidden state in the sequence is assigned to state i as $q_k = i \in S$. The set of duration times is denoted as D; the duration of state i is represented as $d_i \in D$. The transition probability in $\mathbf{A}$ is $a_{(i,d_i)(j,d_j)}$ and it is defined as

$$a_{(i,d_i)(j,d_j)} \quad := \quad P(S_{t+1:t+d_j} = j | S_{t-d_i+1:t} = i). \tag{2.3}$$

Since $a_{(i,d_i)(j,d_j)}$ include the probability of transition and duration together, $v_{i,j}$ is used as simple transition probability distribution from state i to state j and $p(d_j)$ is used as the duration probability of state j separately later. The emission probability in $\mathbf{B}$ is represented as $b_{j,d_j}(\boldsymbol{o}_{t+1:t+d_j})$ and it is defined as

$$b_{j,d_j}(\boldsymbol{o}_{t+1:t+d_j}) \quad := \quad P(\boldsymbol{o}_{t+1:t+d_j} | S_{t+1:t+d_j} = j). \tag{2.4}$$

All notations that is already explained and will be used in the training and recognition phase are summarized in Table 2.2. The set of parameters are updated by the recursive calculation for inference, and the set of updated parameters is represented as $\hat{\Lambda}$.

2.3.4 Model training (inference) in HSMM

This section presents a description of how to train the model of HSMM using training sequences, i.e., how to estimate the set of parameters $\Lambda = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$ including the duration in each state. HSMM is trained using Baum–Welch algorithm [21] in the same way as HMM, where a recursive forward–backward algorithm is used. The forward–backward algorithm is an inference algorithm used for HMM. An extended algorithm special for HSMM is also proposed [79].

Table 2.2: Notations of HSMM.

$o_{1:T}$	the target observation sequence from $t = 1$ to $t = T$
$o_{t_1:t_2}$	the observation sequence from $t = t_1$ to $t = t_2$
Y	set of the observation symbols
y_n	n-th observation symbol in Y
N	the number of symbols in Y
Q	the sequence of hidden states
q_k	k-th hidden state in the hidden state sequence Q
K	the number of hidden states in Q and satisfy $K \leq T$
S	set of the hidden states
M	the number of hidden states in S
i, j	i-th and j-th hidden state in S
$\mathbf{A}$	the transition probability matrix in $\mathbb{R}^{M \times M}$
$\mathbf{B}$	the emission probability matrix in $\mathbb{R}^{M \times N}$
$\boldsymbol{\pi}$	the initial probability vector $\mathbb{R}^{M}$
D	set of the duration time represented as random variable
d_i	the duration of state i in D
$a_{(i,d_i)(j,d_j)}$	the transition probability from state i to state j with duration d_i and d_j in $\mathbf{A}$
a_i	the initial probability of state i in $\boldsymbol{\pi}$
$b_{j,d_j}(o_{t+1:t+d_j})$	the emission probability from state j with duration d_j in $\mathbf{B}$
π_i	i-th element in $\boldsymbol{\pi}$
Λ	set of the parameters $\{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$
$\hat{\Lambda}$	set of the updated parameters $\{\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\boldsymbol{\pi}}\}$
$v_{i,j}(d)$	the transition probability from state i to state j with duration d
$p_{i,j}(d_j)$	the duration probability distribution of state j when transit from state i to state j
$\gamma_k(i)$	the probability that the stochastic process in in state i at k-th state
$\xi_k(i, j)$	the probability that the stochastic process is in state i at k-th state and transits to state j at $k + 1$-th state

The concrete algorithm for HSMM is the following: computing forward probabilities starts from $t = 1$ to $t = T$, with computed backward probabilities from $t = T$ to $t = 1$. This two-way calculation repeats until the likelihood converges. More concretely, the forward step calculates the following forward variable $\alpha_t(j, d_j)$ of state j with d_j at t as

$$\begin{aligned} \alpha_0(i) &= \pi_i b_i(\boldsymbol{o}_0) \\ \alpha_t(j, d_j) &= \sum_{i \in \{S\} \setminus \{j\}} \sum_{d_i \in D} \alpha_{t-d_j}(i, d_i) a_{(i,d_i)(j,d_j)} b_{j,d_j}(\boldsymbol{o}_{t-d_j+1:t}). \end{aligned} \tag{2.5}$$

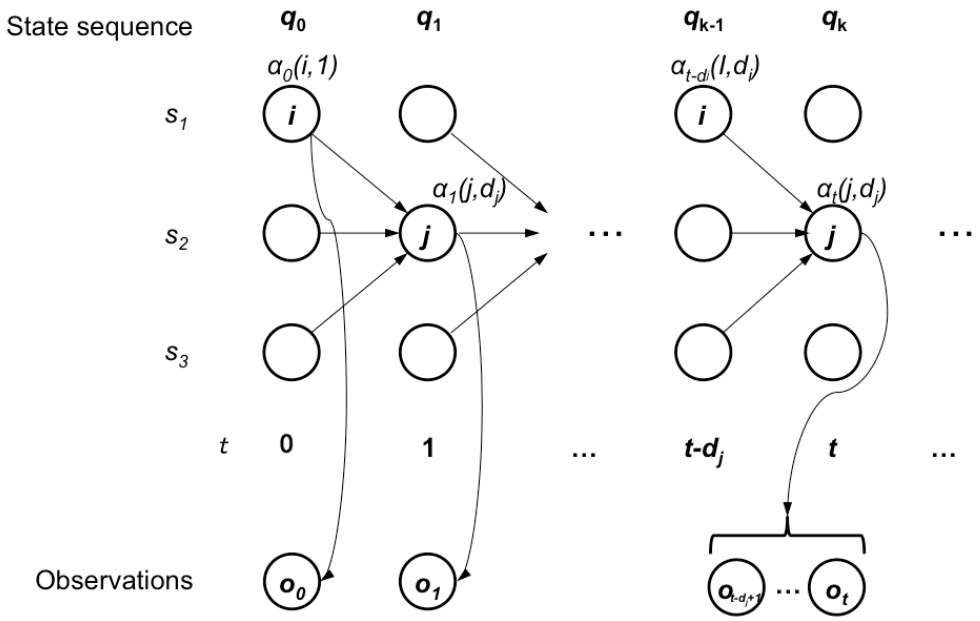


Figure 2.3: Trellis diagram for forward calculation in HSMM.

The trellis diagram for the forward calculation is shown in Figure 2.3. The left area from $t = 1$ to $t = 2$ describes the transition probability from i to j and the duration in state j is $d_j = 1$. The right area from $t = t - d_j$ to $t = t$ describes the transition probability from i to j and the duration in state j is $d_j \neq 1$. It is assumed that the transition probability from state i to state j with d_j does not depend on the length of d_i in general HSMM. The transition probability $a_{(i,d_i)(j,d_j)}$ can be divided into $v_{i,j}p(d_j)$ where $v_{i,j}$ is the transition probability from state i to state j and $p(d_j)$ is the duration probability of state j. The forward variable can be transformed by the following equation

$$\alpha_t(j, d_j) = \sum_{i \in \{S\} \setminus \{j\}} \alpha_{t-d_j}(i, d_i) v_{i,j}(d_j) b_{j,d_j}(\boldsymbol{o}_{t-d_j+1:t}). \tag{2.6}$$

For the simplicity, we use k and $k - 1$ as the index of forward variable instead of t and $t - d_j$ respectively as

$$\alpha_k(j, d_j) = \sum_{i \in \{S\} \setminus \{j\}} \alpha_{k-1}(i, d_i) v_{i,j}(d_j) b_{j,d_j}(\boldsymbol{o}_{t-d_j+1:t}). \tag{2.7}$$

It is used to calculate the sequence likelihood $P(\boldsymbol{o}|\Lambda)$ as follows.

$$P(\boldsymbol{o}|\Lambda) = \sum_{i \ in\{S\}} \alpha_K(i, d_i). \tag{2.8}$$

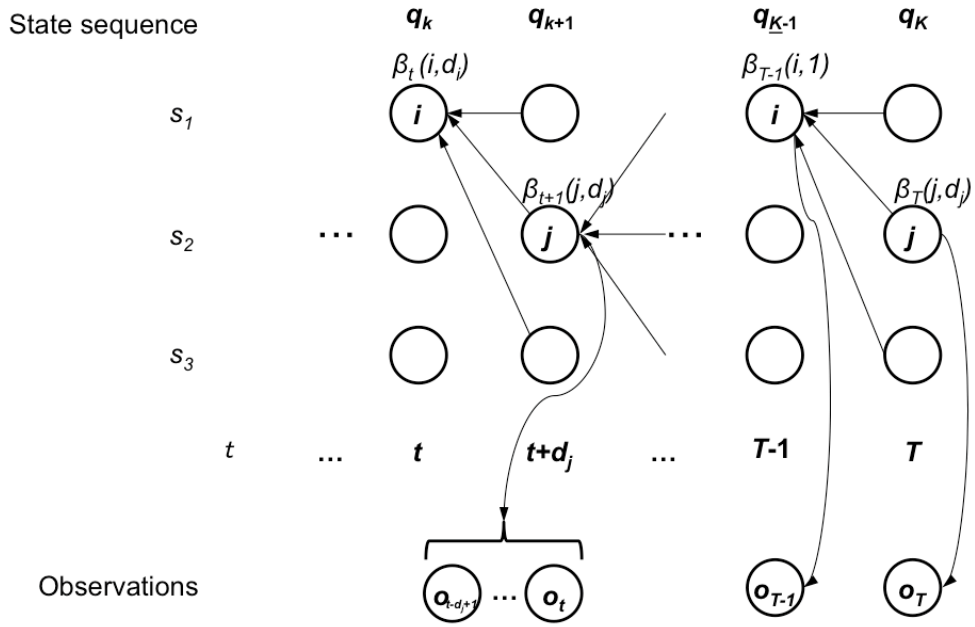


Figure 2.4: Trellis diagram for backward calculation in HSMM.

Then, the backward step calculates the following backward variable $\beta_t(j, d_j)$ as

$$\begin{aligned}\beta_T(j) &= 1 \\ \beta_t(j, d_j) &= \sum_{i \in \{S\} \setminus \{j\}} \sum_{d_i \in D} a_{(j,d_j)(i,d_i)} b_{i,d_i}(\boldsymbol{o}_{t+1:t+d_i}) \beta_{t+d_i}(i, d_i).\end{aligned} \tag{2.9}$$

Likewise the forward variable, the backward variable can be transformed using $v_{i,j}(d_j)$ into the following equation

$$\beta_t(j, d_j) = \sum_{i \in \{S\} \setminus \{j\}} v_{i,j}(d_j) b_{i,d_i}(\boldsymbol{o}_{t+1:t+d_i}) \beta_{t+d_i}(i, d_i). \tag{2.10}$$

The trellis diagram for the backward calculation is shown in Figure 2.4. By using k and $k-1$ as the index of forward variable instead of t and $t - d_j$, the backward variable is replaced as

$$\beta_k(j, d_j) = \sum_{i \in \{S\} \setminus \{j\}} v_{i,j}(d_j) b_{i,d_i}(\boldsymbol{o}_{t+1:t+d_i}) \beta_{k+1}(i, d_i). \tag{2.11}$$

Before representing the re-estimation formula for parameters, γ_k and $\xi_k(i, j)$ are derived. γ_k is the probability that the stochastic process is in the state i at k-th state in the state sequence. It can be calculated using $\alpha_k(i, d_i)$ and $\beta_k(j, d_j)$ as

$$\gamma_k(i) = \frac{\alpha_k(i, d_i)\beta_k(i, d_i)}{\sum_{i \in S} \alpha_k(i, d_i)\beta_k(i, d_i)}. \tag{2.12}$$

Then, $\xi_k(i, j)$ is the transition probability from state i to state j, where k-th state is i and $k+1$-th state is j respectively. The definition of $\xi_k(i, j)$ is

$$\begin{aligned}\xi_k(i, j) &:= P(S_{t-d_i+1:t} = i, q_{t+1:t+d_j} = j | o_{1:T}, \lambda) \\ &:= P(q_k = i, q_{k+1} = j | o_{1:T}, \lambda).\end{aligned} \tag{2.13}$$

It is calculated as

$$\xi_k(i,j) = \frac{\alpha_k(i,d_i)v_{i,j}(d_j)b_{j,d_j}(o_{t+1:t+d_j})\beta_{k+1}(j,d_j)}{\sum_{i\in S}\sum_{j\in S}\alpha_k(i,d_i)v_{i,j}(d_j)b_{j,d_j}(o_{t+1:t+d_j})\beta_{k+1}(j,d_j)}. \tag{2.14}$$

By using the above variables, the re-estimation formula for updated parameters $\hat{\pi}$, $\hat{v}_{i,j}$ and $\hat{b}_{j,d_j}$ can be calculated as following equation. The initial state probability $\hat{\pi}$ is

$$\hat{\pi} = \gamma_0(i). \tag{2.15}$$

The transition probability from state i to state j is updated as

$$\hat{v}_{i,j}(d_j) = \frac{\sum_{k=0}^{K}\xi_k(i,j)}{\sum_{j\in S}\sum_{k=0}^{K}\xi_k(i,j)}. \tag{2.16}$$

The emission probability from state j is updated as

$$\hat{b}_{j,d_j}(y_n) = \frac{\sum_{k=0\ s.t.o_{o_k=y_n}}^{K}\gamma_k(i)}{\sum_{k=0}^{K}\gamma_k(i)}. \tag{2.17}$$

Finally, the probability density distribution of $p(d_j)$ is updated and converged together with $v_{i,j}$. Any probability distribution function can be used as the duration probability distribution. Here, we take the Gaussian distribution for the explanation. The target parameter for the duration probability estimation is represented as θ which is the set of μ and σ in the following gaussian distribution,

$$p(d) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp(-\frac{(d-\mu)^2}{2\sigma^2}). \tag{2.18}$$

The above equation omits the state transition information, but it also can be written as $p_{i,j}(d)$. The objective function for maximization is

$$P(\boldsymbol{d}|(\theta_d)) = \sum_{k=1}^{K}\xi_k(i,j)\log[p_{i,j}(d_k)], \tag{2.19}$$

represented with log-likelihood, where θ_d is the parameter of $p(d)$, i.e., μ and σ. If $\hat{\theta}$ is the updated parameter for the duration distribution function, the parameter is decided when $\hat{\theta} - \theta < \epsilon_d$ where ϵ_d is the predefined threshold for convergence.

The summarized calculation step for estimating model parameters are presented below.

Step 1 Initialization

Give an initial set of parameters Λ of the model at random.

Step 2 Recursive calculation

Calculate the set of parameters $\hat{\Lambda}$ that maximizes the variables of the forward–backward algorithm using the initialized parameter Λ. Denoting the updated state transition probability a and the updated emission probability b as $\hat{a}$ and $\hat{b}$, respectively, $\hat{a}_{(i,d_i)(j,d_j)}$ and $\hat{b}_{j,d_j}(\boldsymbol{o}_{t+1:t+d_j})$ are updated using the previous values of $a_{(i,d_i)(j,d_j)}$ and $b_{j,d_j}(\boldsymbol{o}_{t+1:t+d_j})$. Then, (2.12) and (2.14) are calculated using the variables of (2.5) and (2.9). The parameters dare updated using these equations (2.15), (2.16) and (2.17) as maximizing the sequence likelihood of (2.8).
For each updates, (2.19) is calculated changing θ until it converges in $\hat{\theta} - \theta < \epsilon_d$.

Step 3 Parameter update and log-likelihood calculation

Update the set of parameters as $\Lambda = \hat{\Lambda}$ using the result of **Step 2**. Calculate the probability that outputs the observation sequence $\boldsymbol{o}_{1:T}$ from the current model, and finally calculate the log-likelihood as

$$\hat{\theta} \;=\; \arg\max_{\theta} \log P(\boldsymbol{o}_{1:T}) \;\;=\;\; \log \sum_{j=1}^{M} \alpha_T(j, d_j), \tag{2.20}$$

where $\alpha_T(j, d_j)$ is calculated using (2.5) when $t = T$ at the end of the sequence, and $\hat{\theta}$ is the updated log-likelihood probability.

Step 4 Convergence judgement

Judge whether the estimation process converges by evaluating that the amount of increase from the previous likelihood θ to the updated likelihood $\hat{\theta}$ in **Step 3** is less than a predefined threshold ϵ as

$$\hat{\theta} - \theta \;\;<\;\; \epsilon.$$

If the condition above is satisfied, then the process is terminated. Otherwise **Step 2** and **Step 3** are iterated until the amount of increase converges.

2.3.5 Recognition using HSMM

For the recognition phase that finds the model that is most likely to generate a given target observation sequence, the probability of generating an observation sequence plays a fundamentally important role. For this purpose, we first assume that a *label* is assigned appropriately into each group of sequence in advance. The recognition step is defined to seek the most suitable label for a given group of sequence by calculating the label of the model that has the maximum probability as a recognition result. The probability of generating the target observation sequence is calculated using the forward algorithm used in HMM. For each model, it recursively calculates the forward variable and the probability for each state using $P(\boldsymbol{o}_{1:T}) = \sum_{i=1}^{M} \alpha_T(i, d_i)$, which is the marginal probability distribution, where

$$\alpha_t(j, d_j) = \left[\sum_{i=1}^{M} \alpha_{t-d_j}(i, d_i) a_{(i,d_i)(j,d_j)} \right] b_{j,d_j}(\boldsymbol{o}_{t-d_j+1:t}). \tag{2.21}$$

However, the equation needs calculation for all possible state i in all t. For finding the most probable sequence of states to the target observation effectively, Viterbi algorithm is used. It is similar to the forward variable, but the maximum probability is used instead of all forward variable summation.The objective function is represented as $\delta_k(j, d_j)$. It is defined as

$$\delta_k(i, d_i) := \max_{q_{0:k-1}} P(o_{0:t}, q_1, \cdots q_{k-1}, q_k = i|\Lambda). \tag{2.22}$$

It can be calculated as

$$\begin{aligned} \delta_0(i, d_i) &= \pi_i b_i(o_0) \\ \delta_k(j, d_j) &= \max_{1 \le i \le M} \delta_{k-1}(i) v_{i,j}(d_j) b_{j,d_j}(o_{t+1:t+d_j}). \end{aligned} \tag{2.23}$$

Then, $P(\boldsymbol{o}_{1:T}|\Lambda^z) = \arg\max_i \delta_K(i, d_i)$ is the probability of the state sequence generating the observation sequence $o_{1:T}$ using the parameter set of model z, i.e., Λ^z, i.e., Λ^z, where $z \in \{1, 2, \cdots, Z\}$ and Z are the total number of models. Finally, the label that has the maximum $P(\boldsymbol{o}_{1:T})$ for the observation sequence is selected as the recognition result. Consequently, the model z^* that has the maximum probability $P(\boldsymbol{o}^*_{1:T}|\Lambda^z)$ among all Z models is selected as a result of the recognition.

Algorithm 1 Algorithm for training and recognition in HSMM.

Require: Input
Training sequences: $\boldsymbol{o}^z_{1:T_r} = \{o^z_1, \cdots, o^z_{T_r}\}$,
Testing sequences: $\boldsymbol{o}^*_{1:T_t} = \{o^*_1, \cdots, o^*_{T_t}\}$.
(Z is the number of training sequences.)
(H is the number of recursive calculation.)
Ensure: Training phase
for $z = 1$ to Z **do**
  Assign random values to the HSMM parameters $\Lambda^z = \{A, B, \pi\}$, and $\alpha_{t(j,d_j)}$ and $\beta_{t(j,d_j)}$.
  for $h = 1$ to H **do**
    for $t = 1$ to T_r **do**
      Calculate $\alpha_{t(j,d_j)}$ and $\beta_{t(j,d_j)}$ using (2.5) and (2.9).
      Update parameters Λ^z using (2.16) and (2.17) as maximizing the likelihood (2.8).
    end for
    Calculate θ_h using (2.20).
    if $\theta_h - \theta_{h-1} < \epsilon$ **then**
      break
    end if
  end for
end for
Ensure: Recognition phase
for $z = 1$ to Z **do**
  for $t = 1$ to T_t **do**
    Prepare Λ^z from the results obtained in the training phase.
    Calculate $\alpha_t(j, d_j)$ using (2.21).
  end for
  Calculate $P(o_{1:T_t}|\Lambda^z)$ using $\alpha_t(j, d_j)$.
end for
Select the model z^* that has the maximum value for $P(o^*_{1:T_t}|\Lambda^z)$.
Return Model z^* and its probability $P(\boldsymbol{o}^*_{1:T_t}|\Lambda^{z^*})$.

Chapter 3

Duration and interval modeling with HSMM

3.1 State interval modeling in HSMM

This section presents investigation of how to model a state interval in a model using HSMM. Before explaining the details, we describe how to represent state interval in a sequence. The baseline HSMM model ignores the period when no event is observed because the occurrence of events and the order of the events are necessary for sequential data modeling. However, we also consider how to deal with the no-observation period to satisfy the requirement of "interval between events" as described in Section 2.3.1. Therefore, we regard this period as the state interval in this paper, and assign a new symbol "interval symbol" to this period. Figure 3.1 portrays an example of the state interval representation, where "**a**" and "**b**" are symbols that are actually observed in the original sequence, and "**i**" is the interval symbol used to fill the state interval. Section 3.1.1 examines the approaches for modeling state interval using HSMM. The issues that arise because of the filled sequence with state interval are addressed in Section 3.1.2.

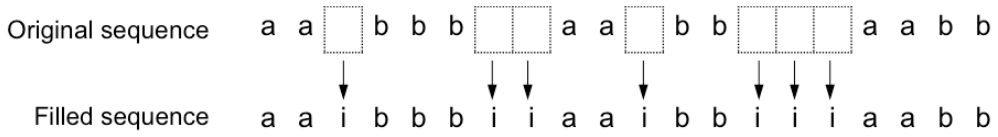


Figure 3.1: Representation of state interval in a sequence.

3.1.1 Two approaches for state interval modeling

To treat state interval with HSMM, two approaches can be regarded as shown in Figure 3.2. One represents state interval as a new probabilistic parameter as shown in Figure 3.2(a). The other approach represents the state interval as a new state node, which is represented as a black node as Figure 3.2(b). Each state of HSMM can represent its duration for staying in a single state. Therefore, this new approach describes the length of the state interval by introducing the new state node that explicitly indicates the state interval.

For both approaches, three variations to model the state interval can be considered. The first approach models the state interval with the preceding state ((a)-1, (b)-1); the second models it

with the subsequent state ((a)-2, (b)-2). The last variation models the length of the interval with both preceding and subsequent states ((a)-3, (b)-3). Compared among three variations, the first two models have connection with only one state whereas the last one ((a)-3, (b)-3) has connections with two states. Therefore, (a)-3 and (b)-3 can model the sequential data more precisely.

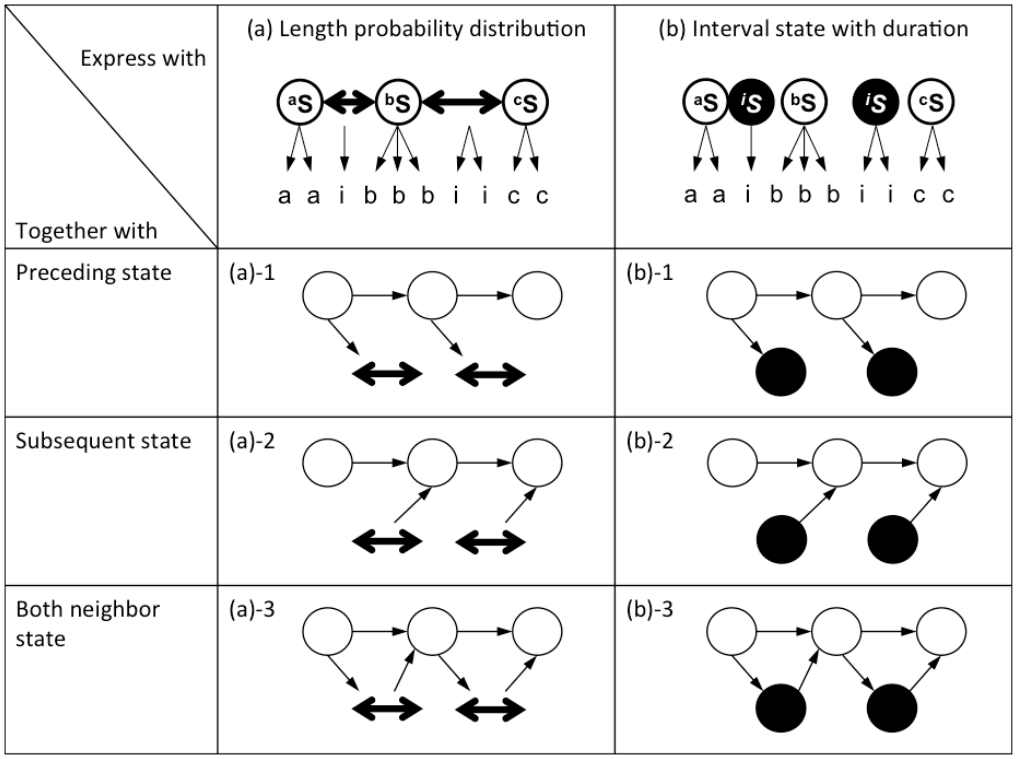


Figure 3.2: Two approaches for the state interval. The circle represents a state and $\rightarrow$ represents the transition from the left state to the right state. Circle filled with black and $\leftrightarrow$ represent the state interval.

3.1.2 Problems of state interval modeling

Before describing the proposed models, the technical issues for the state interval modeling in each approach in the preceding subsection are explained. For the first approach in the preceding subsection, the manner of representing a state interval with the new probabilistic parameter "interval length probability" must be defined. Considering the application data, the model is expected to be found such that sequential data have a similar sequential pattern with similar state duration and a state interval. Therefore, it is necessary to model the state duration and state interval with representation of the similarity of its time length. Therefore, the second approach defines how to represent the new parameter for state duration and how to model the parameters with original HSMM in a probabilistic manner.

For the second approach, the following problems occur. The structure of the second approach is presented in Figure 3.3, where the interval state node is presented as a black node ${}^{i}S$. Although this approach handles the state interval in a simple way, it causes large bias in the transition probability when there are many groups of terms of observed interval symbols in a sequence as shown in Figure 3.4. Figure 3.4(a) presents an example sequence for the explanation. Each sequence shows

the original observation sequence and the state sequence. Figure 3.4(b) presents an example sequence filled with state interval nodes of interval symbol **i**. The tables represented at the right of the figure show the transition frequency from a state to another state calculated using the original/complemented sequence. Whereas the states described in a vertical line in the table show the "from" states, the states in a horizontal line show the "to" state. The table in (a) shows the transition frequency calculated using the original state sequence. The table in (b) shows the transition frequency calculated using the converted state sequence filled with interval states. Accordingly, the results reveal that the transition frequency in the whereas cells except for gray painted cells falls dramatically to lower level, i.e., nearly zero. This means that, the introduction of the interval state node causes a deviation to the original transition probability. The resultant new model fails to represent the transition sequence properly.

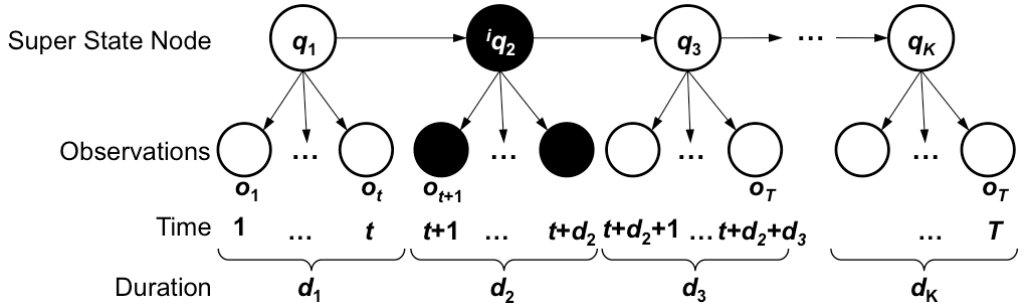
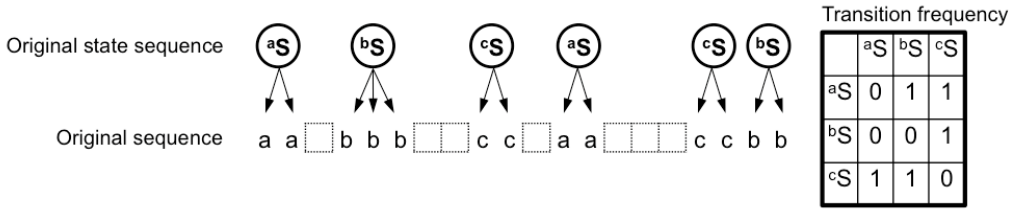
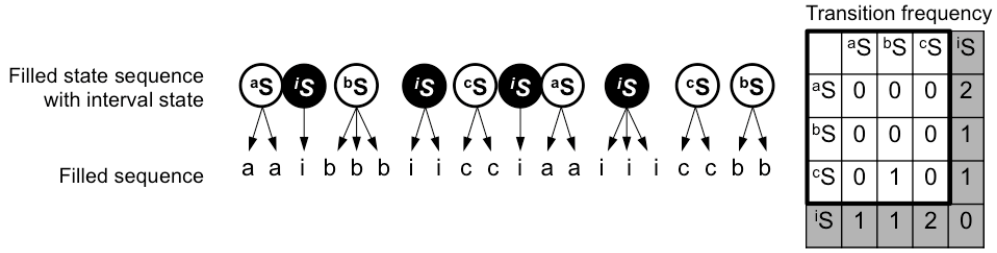


Figure 3.3: HSMM with an interval state.



(a) Original sequence and transition frequency.



(b) Filled sequence and transition frequency.

Figure 3.4: Problem of sequence with a state interval.

Addressing these problems, finally, we propose two models in the following sections: an interval length probability hidden semi-Markov model (ILP-HSMM) for the first approach, and an interval state hidden semi-Markov model (IS-HSMM) as the second approach.

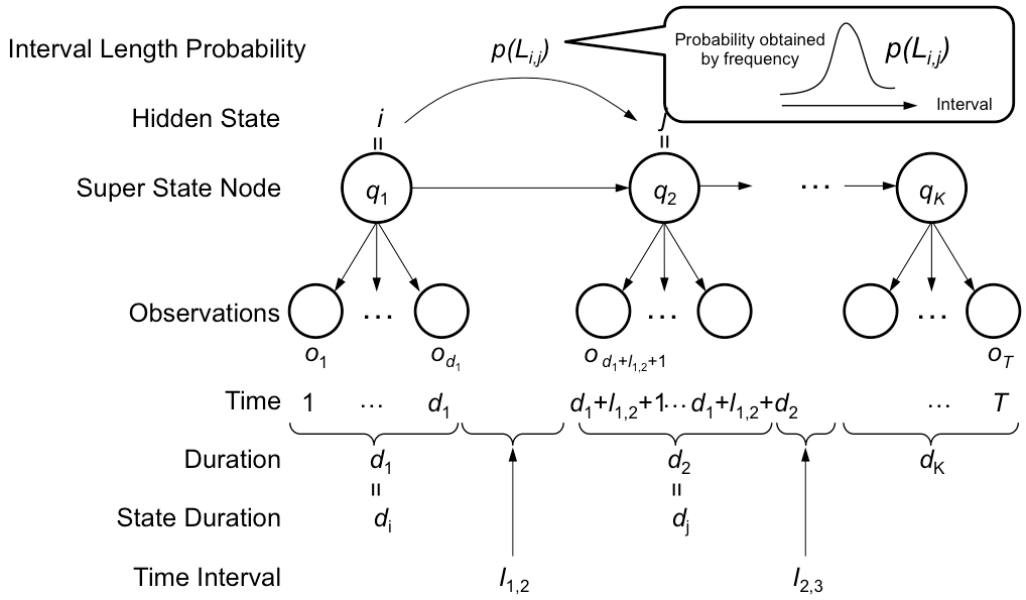


Figure 3.5: Conceptual structure of ILP-HSMM using state interval probability.

3.2 Interval Length Probability HSMM (ILP-HSMM)

This section presents ILP-HSMM, which newly introduces interval length probability to the transition probability to handle the state interval between two states. It is noteworthy that the interval length probability corresponds to the probability density distribution of interval length of two states, to be technically precise. The distinct difference between HSMM and ILP-HSMM is that, whereas state j starts immediately after the end time of state i in the original HSMM, state j starts after a length of time, $L_{i,j}$, passes since the end time of state i in ILP-HSMM. The training phase deals with $L_{i,j}$ as skipped time and trains the sequences removed the interval as the general HSMM and also trains the interval length probability from state i to state j respectively. It is noteworthy that the total time length of the observation sequence T varies because of its dependency on the length of state duration and interval, leading to $T = \Sigma_{k=1}^{K}(d_k + l_{k-1,k})$, where $l_{k-1,k}$ is the time difference between the end of q_{k-1} and the beginning of q_k. The subsequence section presents a description of how to model the probability of interval length and how to recognize using the interval length probability given datasets using ILP-HSMM.

3.2.1 Model training (inference) in ILP-HSMM

Figure 3.6 presents example data and representations used hereinafter for explanation. The slash line patterned blocks represent the data sequence of the training dataset. The set of parameters used in ILP-HSMM is defined as

$$\Lambda := \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}, \mathbf{L}\},$$

where the elements of the parameter Λ take on $\mathbf{A}(i,j) = a_{(i,d_i)(j,d_j)}$, $\mathbf{B}(j,n) = b_{(j,d_j)}(\boldsymbol{o}_{1:d_j})$, and $\boldsymbol{\pi}(i) = \pi_{j,d_j}$, where $d_i \in D$ represents the duration of state i described in Section 2.3.3. Furthermore, $\mathbf{L} \in \mathbb{R}^{M \times M}$ is the matrix that consists of the interval length probabilities, i.e., the probability density distributions of state interval length, where $\mathbf{L}(i,j) = p(L_{i,j})$. The transition and emission

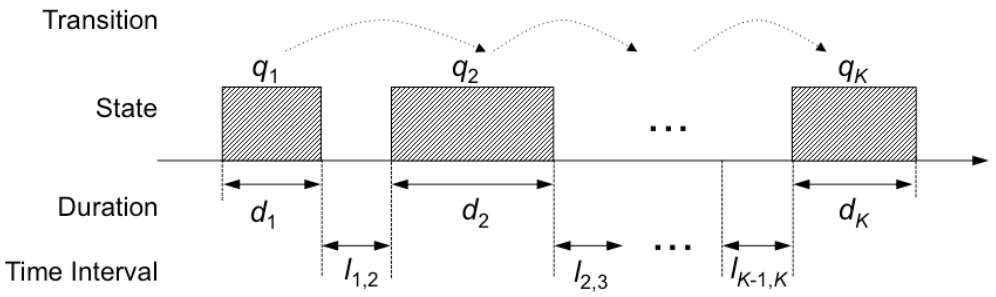


Figure 3.6: Sequential data and representations.

probabilities are defined as the same as those in HSMM. The difference between HSMM and ILP-HSMM is to consider the parameter of $p(L_{i,j})$.

Then, the probability density distribution of the interval length of $L_{i,j}$ is expressed by the Gaussian distribution $p(L_{i,j})$ as

$$p(L_{i,j}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(L_{i,j}-\mu)^2}{2\sigma^2}}, \tag{3.1}$$

where σ and μ respectively present the variance and the mean of $L_{i,j}$. It is noteworthy that the Gaussian distribution is adopted as the probability density distribution, for simplicity. However, other distributions and functions were adopted for ILP-HSMM without changing any other parameter.

Here, we show the example of Gaussian distribution for the duration distribution, it can be replaced with other kernels according to the dataset, for example, log-normal distribution, binomial distribution, Poisson distribution and so on. Binomial distribution is not density distribution, and these approximations are Gaussian distribution or Poisson distribution. In case that the penalties for duration difference are required the same probability for positive and negative direction, the Gaussian distribution is suitable. On the other hand, the penalties are required with bias depends on the elapsed time, other Poisson or log-normal distribution are suitable. The simple discrete probability distribution that is represented as the normalized probability of occurrence frequency can be used as the duration probability. It is not the parametric, but the probability is suitable in case that it does not allow the small gap of the interval length. This model train the interval length probability independent of the other parameters, it can be applied other distribution functions by simply replacing the kernel.

The range of $L_{i,j}$ in (3.1) might influence either memory consumption or computational complexity to generate the model. There might be no $L_{i,j}$ value suitable for the observation values because of the range limitation of $L_{i,j}$ if $p(L_{i,j})$ is generated in a training period. However, if the parameter $p(L_{i,j})$ is generated every time an observation is fed to the algorithm, then the calculation cost can be much higher. Our motivation to introduce the interval length probability to HSMM is, as explained earlier, to find the similar part of sequential data with respect to the state interval and also to discriminate between the target part and the similar part. Therefore, even if the probability of $L_{i,j}$ is presumed to be zero around the skirts of the distribution, no critically important difficulty arises. Consequently, we introduce the boundary of the probability value δ_{pt} to ascertain the edge of the skirt of $p(L_{i,j})$. On generating the $p(L_{i,j})$, the calculation is terminated when the probability value becomes less than δ_{pt}. The probability of $p(L_{i,j})$ is zero outside of the range of δ_{pt}.

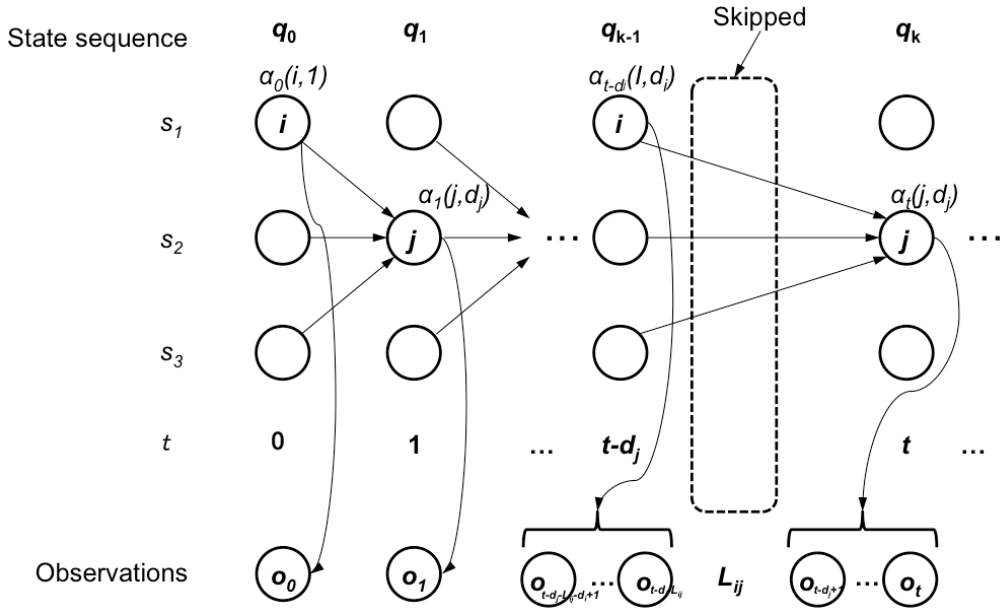


Figure 3.7: Trellis diagram for forward calculation in ILP-HSMM.

3.2.2 Recognition using ILP-HSMM

The interval length probability trained in the training phase is used in the recognition phase. The Viterbi algorithm is used to estimate the probability of a model [55]. The pair of the model with the interval length probability and its label that is expected to be estimated are stored as candidates for estimation. The recognition label that denotes the estimated result is selected when the model has the maximum likelihood estimate by calculating it for each state in each model.

First, we calculate $p(L_{i,j})$ beforehand. The forward variable is defined as

$$\begin{aligned}\alpha_0(i) &= \pi_i b_i(\boldsymbol{o}_0) \\ \alpha_t(j,d_j) &= \sum_{i\in\{S\}\backslash\{j\}} \sum_{d_i\in D} \alpha_{t-d_j}(i,d_i) a_{(i,d_i)(j,d_j)} b_{j,d_j}(\boldsymbol{o}_{t-d_j+1:t}) p(L_{i,j}).\end{aligned} \tag{3.2}$$

It is calculated using k as

$$\alpha_k(j,d_j) = \sum_{i\in\{S\}\backslash\{j\}} \alpha_{k-1}(i,d_i) v_{i,j}(d_j) p(L_{i,j}) b_{j,d_j}(\boldsymbol{o}_{t-d_j+1:t}). \tag{3.3}$$

Then, the forward variable for estimating the maximum likelihood is calculated as

$$\alpha_t(j,d_j) = \left[\sum_{i=1}^{M} \alpha_{t-d_j}(i,d_i) a_{(i,d_i)(j,d_j)} p(L_{i,j})\right] b_{j,d_j}(\boldsymbol{o}_{t-d_j+1:t}), \tag{3.4}$$

and the likelihood when reached at T is calculated as $P(\boldsymbol{o}_{1:T}) = \sum_{i=1}^{M} \alpha_T(i,d_i)$.

Finally, the probability that the model generate the observation sequence is calculated as $P(\boldsymbol{o}_{1:T}|\Lambda^z) = \arg\max_i \delta_K(i,d_i)$ where

$$\begin{aligned}\delta_0(i,d_i) &= \pi_i b_i(o_0) \\ \delta_k(j,d_j) &= \max_{1\le i\le M} \delta_{k-1}(i) v_{i,j}(d_j) p(L_{i,j}) b_{j,d_j}(o_{t+1:t+d_j}).\end{aligned} \tag{3.5}$$

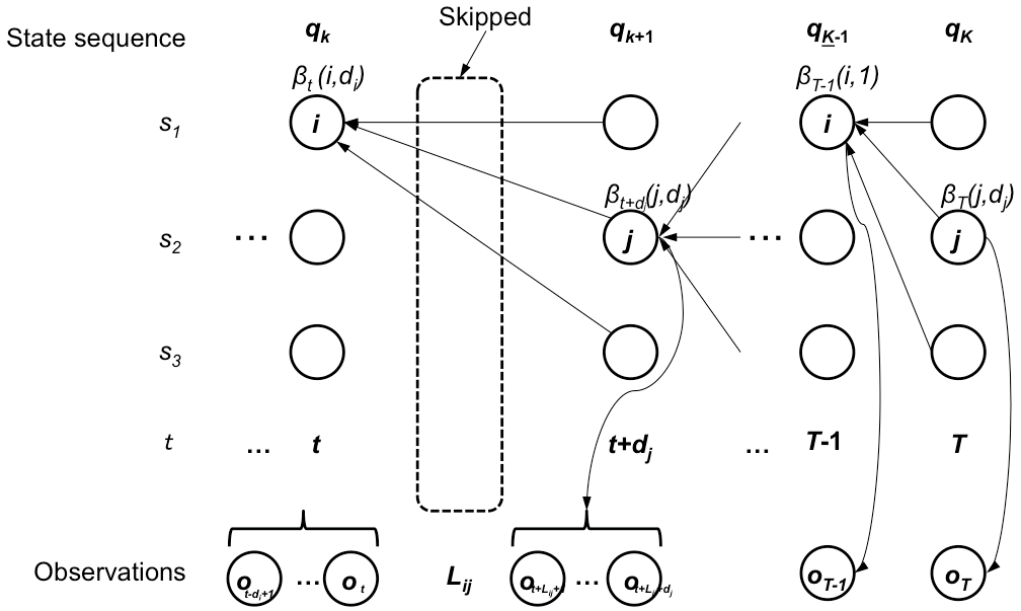


Figure 3.8: Trellis diagram for backward calculation in ILP-HSMM.

The interval length probability is calculated simultaneously with calculation of the parameter of the likelihood using the transition probability recursively.

The difference between HSMM and ILP-HSMM is the capability of handling the length of the state interval between states as explained earlier. The interval length probability in ILP-HSMM can be integrated by introducing each interval into two pair of states to calculate the likelihood. This calculation might produce an additional calculation cost. Therefore, it is necessary to evaluate the calculation cost. In addition, the emission probability $b_{j,d_j}(\mathbf{o}_{1:d_j})$ can be parametric or non-parametric. In this proposal, the relation between the state duration and state interval is not represented in a model. For this reason, $b_{j,d_j}(\mathbf{o}_{1:d_j})$ is handled as non-parametric, discrete, and independent of the duration. Then, $p(L_{i,j})$ is also discrete and independent of the duration and the transition probability.

The overall algorithm is presented in Algorithm 2.

Algorithm 2 Algorithm for training and recognition in ILP-HSMM.

Require: Input
Training sequences: $\boldsymbol{o}^z_{1:T_r} = \{o^z_1, \cdots, o^z_{T_r}\}$,
Testing sequences: $\boldsymbol{o}^*_{1:T_t} = \{o^*_1, \cdots, o^*_{T_t}\}$.
(Z is the number of training sequences.)
(H is the number of recursive calculation.)
Ensure: Training phase
for $z = 1$ to Z **do**
  Assign random values to the HSMM parameters $\Lambda = \{A, B, \pi, L\}$, and $\alpha_{t(j,d_j)}$ and $\beta_{t(j,d_j)}$.
  Initialize $p(L_{i,j})$ as $L_{i,j} = 1$.
  for $h = 1$ to H **do**
    for $t = 1$ to T_r **do**
      Calculate $\alpha_{t(j,d_j)}$ and $\beta_{t(j,d_j)}$ using (2.5) and (2.9).
      Calculate $p(L_{i,j})$ with i and j using (3.1).
      Update parameters Λ.
    end for
    Calculate θ_h using (3.16).
    if $\theta_h - \theta_{h-1} < \epsilon$ **then**
      break
    end if
  end for
end for
Ensure: Testing phase
for $z = 1$ to Z **do**
  for $t = 1$ to T_t **do**
    $\Lambda^z \leftarrow$ parameter Λ of model z.
    $p(l) \leftarrow p(L_{i,j})$ using Λ^z with observed interval l.
    Calculate $\alpha_t(j, d_j)$ using (2.21) with (3.4).
  end for
  Calculate $P(o_{1:T_t}|\Lambda^z)$ using $\alpha_t(j, d_j)$.
end for
Select model z^* with the maximum value for $P(\boldsymbol{o}^*_{1:T_t}|\Lambda^z)$.
Return Model z^* and its probability $P(\boldsymbol{o}^*_{1:T}|\Lambda^{z^*})$.

3.3 Interval State Hidden Semi-Markov Model (IS-HSMM)

Actually, HSMM handles the state interval in a simple way because the interval symbol is replaced with the new interval state node as described in Section 3.1. However, we face the difficulty of the degradation of the accuracy of the transition probability in cases where state intervals appear frequently in the same sequence. To resolve this difficulty, we propose an extended model, IS-HSMM, to preserve the transition probability of the original sequence. Figure 3.9 presents a conceptual structure of IS-HSMM. For easy-to-understand explanation, we select the first three states shown in Figure 3.9 as an example when q_1 and q_3 are original hidden states and ${}^{i}q_2$ is the interval state node. Whereas the original HSMM infers the transition probability in the order of q_1, ${}^{i}q_2$, and q_3, the proposed IS-HSMM infers the transition probability as q_3 using two transition probabilities not only from ${}^{i}q_2$ to q_3, but also from the previous q_1 to q_3 to preserve the transition of the original sequence. This is a noteworthy feature of IS-HSMM. This section explains how to train and how to recognize the model as follows.

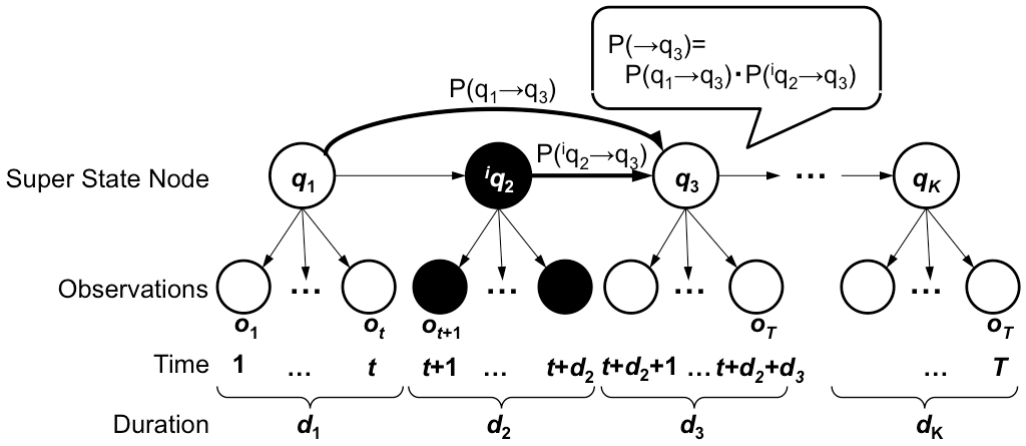


Figure 3.9: Conceptual structure of IS-HSMM with an interval state node using two transition probabilities.

3.3.1 Model training (inference) in IS-HSMM

The difference against the baseline HSMM model appears in the calculation of the forward variables and backward variables in the recursive calculation step. The state transition probability from state i to state j, where the interval state ${}^{i}s$ is inserted between state i and state j, is defined as

$$\begin{aligned} a_{(i,d_i)({}^{i}s,{}^{i}d)(j,d_j)} &:= P(S_{t+{}^{i}d+1:t+{}^{i}d+d_j} = j | S_{t+1:t+{}^{i}d} =^{i} s,\ S_{t-d_i+1:t} = i) \\ &:= P(S_{t+1:t+d_j} = j | S_{t-{}^{i}d+1:t} =^{i} s,\ S_{t-d_i-d_i+1:t-{}^{i}d} = i) \end{aligned}$$

where the duration of interval state ${}^{i}s$ is denoted as ${}^{i}d(>0)$. The respective durations of state i and j are d_i and d_j. The transition $a_{(i,d_i)({}^{i}s,{}^{i}d)(j,d_j)}$ is calculated with the transition from ${}^{i}s$ and the preceding state s_i only when calculating after ${}^{i}s$. Therefore, the forward variable, where the current state is j and the preceding state is ${}^{i}s$, is calculated using the further preceding state i based on the second-order HMM [80] as

$$\alpha_t(({}^{i}s,{}^{i}d),(j,d_j)) = \sum_{i\in\{S\}\setminus\{j,{}^{i}s\}} \sum_{d_i\in D} \alpha_{t-{}^{i}d}((i,d_i),({}^{i}s,{}^{i}d)) \cdot a_{(i,d_i)({}^{i}s,{}^{i}d)(j,d_j)} b_{j,d_j}(o_{t-d_j+1:t}). \tag{3.6}$$

The forward variable calculation equation is represented using k when the previous state if ${}^{i}s$ as

$$\alpha_k(j, d_j) = \sum_{i \in \{S\} \backslash \{j\}} \alpha_{k-2}(i, d_i) v_{i,{}^{i}s}({}^{i}d) v_{{}^{i}s,j}(d_j) b_{j,d_j}(\boldsymbol{o}_{t-d_j+1:t}). \tag{3.7}$$

The sequence likelihood is calculated as

$$P(\boldsymbol{o}|\Lambda) = \sum_{i \ in\{S\}} \alpha_K(i, d_i), \tag{3.8}$$

that is the same calculation with (2.8).

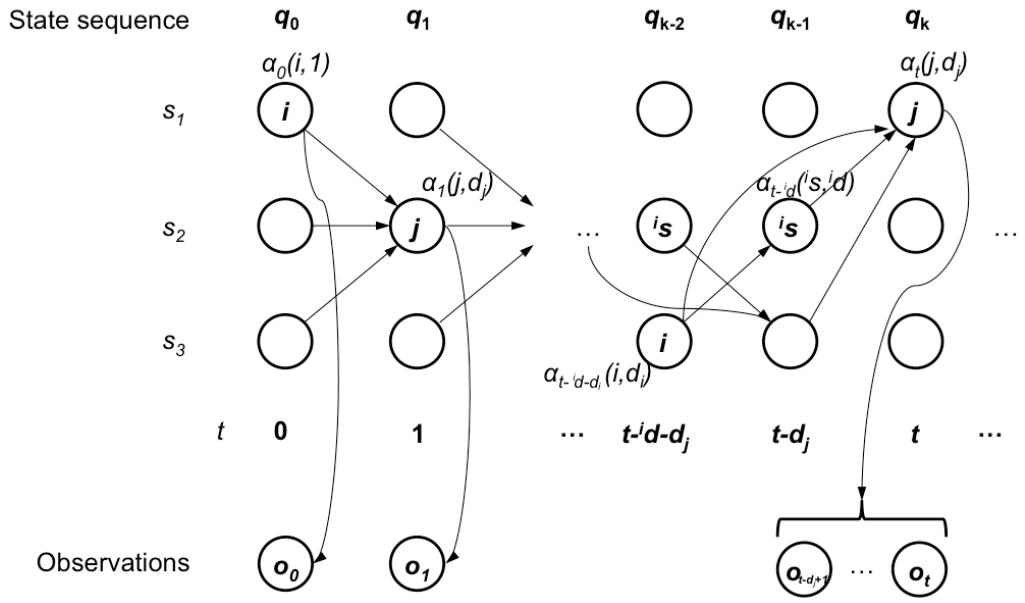


Figure 3.10: Trellis diagram for forward calculation in IS-HSMM.

Trellis diagram for the forward calculation is shown in Figure 3.10. The left area from $t = 1$ to $t = 2$ describes the transition probability from i to j and the duration in state j is $d_j = 1$. A part of IS-HSMM is the general HSMM in case of no ${}^{i}s$ exist. The right area from $t = t - d_j$ to $t = t$ describes the transition probability from i to j via ${}^{i}s$ and the duration in state ${}^{i}s$ and j is ${}^{i}d$ and d_j respectively. When the previous state is ${}^{i}s$, the further previous forward variable $\alpha_{t-{}^{i}d-d_j}(i, d_i)$ is used to calculate $\alpha_t(j, d_j)$. If the state ${}^{i}s$ is identified *a priori*, it is not necessary to be estimated as semi-supervised learning. However, if the state ${}^{i}s$ is not identified, it is necessary to be estimated which state is ${}^{i}s$. The target state to be identified is that the number of transitions to the state and from the state is remarkably higher than other states as described in Figure 3.4. Therefore, we estimate the parameters $\hat{\Lambda}$ using the forward-backward algorithm in HSMM, and then identify the interval state ${}^{i}s$ using the following equation.

$$\hat{i} = \arg\max_i \left(\frac{1}{M-1} \sum_{j \in S \backslash \{i\}} a_{i,j}\right). \tag{3.9}$$

where $\hat{i}$ is the candidate of ${}^{i}s$. Since the existence of the interval state ${}^{i}s$ is assumed, ${}^{i}s$ is specified as ${}^{i}s = \hat{i}$. If there is no such assumption, it can be judged whether the model treat $\hat{i}$ as ${}^{i}s$ by the

following determination formula. If the state $\hat{i}$ satisfies

$$\sum_{x \in S}(x-\mu)^2 - \sum_{x \in S \setminus \{\hat{i}\}}(x-\mu)^2 > \epsilon, \tag{3.10}$$

$\hat{i}$ is treated as ${}^{i}s$. μ is the mean value of the transition probabilities from any i to j calculated as

$$\mu = \frac{1}{M-1}\sum_{j \in S \setminus \{\hat{i}\}} a_{i,j}. \tag{3.11}$$

If the state satisfies the following equation, it is identified as the interval state ${}^{i}s$. This calculation is useful not only for the interval state but also the state which cause the bias for the transition probability in a model.

Then, the backward variable where the preceding state is ${}^{i}s$ is calculated as general first-order transition probabilities is expressed as

$$\beta_t(j, d_j) = \sum_{i \in \{S\} \setminus \{j\}} \sum_{{}^{i}d \in D} a_{(j,d_j)({}^{i}s,{}^{i}d)} b_{{}^{i}s,{}^{i}d}(\boldsymbol{o}_{t+1:t+{}^{i}d}) \beta_{t+{}^{i}d}({}^{i}s,{}^{i}d). \tag{3.12}$$

The backward variable when the previous state is ${}^{i}s$ can be transformed using $v_{i,j}(d_j)$, and k as the index of the backward variable into the following equation

$$\beta_k(j, d_j) = \sum_{i \in \{S\} \setminus \{j\}} v_{i,{}^{i}s}({}^{i}d) v_{{}^{i}s,j}(d_j) b_{i,d_i}(\boldsymbol{o}_{t+1:t+d_i}) \beta_{k+1}(i, d_i). \tag{3.13}$$

Trellis diagram for backward calculation is described in Figure 3.11. The probability of the stochastic

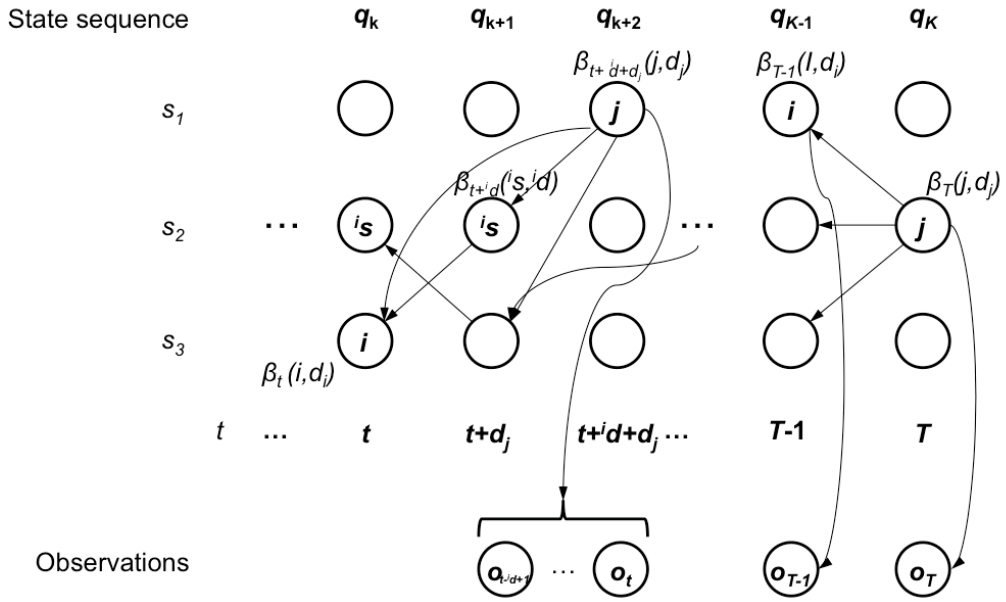


Figure 3.11: Trellis diagram for backward calculation in IS-HSMM.

process at state i to state j via ${}^{i}s$ where starting from k is

$$\begin{aligned}\xi_k(i,{}^{i}s,j) &:= P(S_{t-d_i+1:t} = i, S_{t+1:t+{}^{i}d} =^{i} s, q_{t+{}^{i}d+1:t+{}^{i}d+d_j} = j | o_{1:T}, \lambda) \\ &:= P(q_k = i, q_{k+1} =^{i} s, q_{k+2} = j | o_{1:T}, \lambda).\end{aligned} \tag{3.14}$$

It is calculated as

$$\xi_k(i,{}^i s, j) = \frac{\alpha_k(i,d_i)v_{i,{}^i s}({}^i d)v_{{}^i s,j}(d_j)b_{j,d_j}(o_{t+{}^i s+1:t+{}^i s+d_j})\beta_{k+2}(j,d_j)}{\sum_{i\in S}\sum_{j\in S}\alpha_k(i,d_i)v_{i,{}^i s}({}^i d)v_{i,j}(d_j)b_{j,d_j}(o_{t+{}^i s+1:t+{}^i s+d_j})\beta_{k+1}(j,d_j)}. \tag{3.15}$$

Finally, the transition probability and the emission probability are updated using (3.6) and (3.12) by calculating the state transition probability using (3.6) and assigning the forward and backward variables obtained respectively using (2.16) and (2.17).

3.3.2 Recognition using IS-HSMM

Although calculation of the probability follows the original HSMM when the preceding state is not the interval state node, it differs when the preceding state is the interval state node. The probability of the observation sequence when the preceding state is the interval state node is calculated as $P(\boldsymbol{o}_{1:T}) = \sum_{i=1}^{M}\alpha_T(i,d_i)$, where (2.21) and the follows:

$$\alpha_t(j,d_j) = \left[\sum_{i=1}^{M}\alpha_{t-d_j-{}^i d}(i,d_i)a_{(i,d_i)({}^i s,{}^i d)}\cdot a_{({}^i s,{}^i d)(j,d_j)}\right] b_{{}^i s,{}^i d}(\boldsymbol{o}_{t-{}^i d-d_j+1:t-d_j})\cdot b_{j,d_j}(\boldsymbol{o}_{t-d_j+1:t}) \tag{3.16}$$

where the preceding state is ${}^i s$. It is also replaced using k as

$$\begin{aligned}
\delta_0(i,d_i) &= \pi_i b_i(o_0)\\
\delta_k(j,d_j) &= \max_{1\le i\le M\setminus\{{}^i s\}}\delta_{k-1}(i)v_{i,j}(d_j)b_{j,d_j}(o_{t+1:t+d_j})\\
\delta_k(j,d_j) &= \max_{{}^i s}\delta_{k-2}(i)v_{i,{}^i s}({}^i d)v_{{}^i s,j}(d_j)b_{j,d_j}(o_{t+{}^i s+1:t+{}^i s+d_j}).
\end{aligned} \tag{3.17}$$

The overall algorithm is presented in Algorithm 3.

Algorithm 3 Algorithm for training and recognition in IS-HSMM.

Require: Input
Training sequences: $\boldsymbol{o}^z_{1:T_r} = \{o^z_1, \cdots, o^z_{T_r}\}$,
Testing sequences: $\boldsymbol{o}^*_{1:T_t} = \{o^*_1, \cdots, o^*_{T_t}\}$.
(Z is the number of training sequences.)
(H is the number of recursive calculation.)
Ensure: Training phase
for $z = 1$ to Z **do**
  Assign random values to the HSMM parameters $\Lambda = \{A, B, \pi\}$, and $\alpha_{t(j,d_j)}$ and $\beta_{t(j,d_j)}$.
  for $h = 1$ to H **do**
    for $t = 1$ to T_r **do**
      if o_{t-1} is interval symbol **then**
        Calculate $\alpha_{t(j,d_j)}$ and $\beta_{t(j,d_j)}$ with joint probability from i and ${}^is$ using (3.6) and (3.12).
      else
        Calculate $\alpha_{t(j,d)}$ and $\beta_{t(j,d_j)}$ with preceding state i using (2.5) and (2.9).
      end if
      Update parameters Λ.
    end for
    Calculate θ_h using (2.20) with (3.6).
    if $\theta_h - \theta_{h-1} < \epsilon$ **then**
      break
    end if
  end for
end for
Ensure: Testing phase
for $z = 1$ to Z **do**
  for $t = 1$ to T_t **do**
    if o_{t-1} is the interval symbol **then**
      $\Lambda^z \leftarrow$ parameter Λ of model z with joint probability from j and ${}^is$.
    else
      $\Lambda^z \leftarrow$ parameter Λ of model z with preceding state j.
    end if
    Calculate $\alpha_t(j, d_j)$ using (2.21) with (3.16).
  end for
  Calculate $P(o_{1:T_t}|\Lambda^z)$ using $\alpha_t(j, d_j)$.
end for
Select the model z^* that has the maximum value for $P(\boldsymbol{o}^*_{1:T_t}|\Lambda^z)$.
Return Model z^* and its probability $P(\boldsymbol{o}^*_{1:T_t}|\Lambda^{z^*})$.

3.4 Evaluations

This section presents a description of the performance evaluation of models. After explaining the specifications of the experimental data in Section 3.4.1, Section 3.4.2 and Section 3.4.3 present the experimentally obtained results of the execution time and recognition performance comparison among HSMM, IS-HSMM, and ILP-HSMM. Finally, we evaluate a reproducibility comparison between IS-HSMM and ILP-HSMM in terms of the modeling performance in Section 3.4.4.

3.4.1 Experimental data

Addressing that the sequential data contain the state duration and state interval, we use music sound data played by instruments of different kinds. When the same music is played by the different instruments, even if the music rhythm is the same, the length of each sound for the same note differs. For example, the sound power spectrum played by an organ and drum for the same music sound data is shown in Figure 3.12. The horizontal axis shows the time. The vertical axis shows the sound power, i.e., sound volume. Whereas the power of each note played by the organ is almost identical during the sound resonation, the one played by the drum decreases rapidly after tapping. We generate the observation sequence from the music sound data. The generation step is described below using the features of sound continuous time.

Step 1

Set thresholds b_1 and b_2 to classify the observation symbols into three types by the level of the volume. b_1 is a threshold for determining whether the sound is "on" or "off", and b_2 is the one for classifying the power of the sound as "high" or "low". $(b_2 \geq b_1)$

Step 2

For the sound power v of each time, the observation sequence is generated as follows.

If $v \geq b_2$, then the observation symbol is "high".

If $b_2 > v \geq b_1$, then the observation symbol is "low".

An example of observation sequence generated by the procedure described above is shown in Figure 3.13. The black cell represents the "high" symbol. The gray cell represents a "low" symbol. The white-painted cells represent the "interval." To denote the segment of a sequence, we add "start" and "end" symbols to each edge of the sequence. These symbols are useful for modeling the transition from the initial state from sequences precisely. The dataset consists of 27 segmented data, which are divided into bars of the music sequence. A label is assigned for each 27 segmented data. Therefore the number of labels is also 27. The kinds of the instruments are a grand piano, horn, drums, acoustic guitar, flute, and pipe organ. We use the music sound data played by the instruments of the first three kinds for training data, whereas the latter three kinds are used for recognition data. The numbers of the sequential data are 81 for both training and recognition.

3.4.2 Execution time evaluation

This section presents the execution time evaluation for training and recognition. For the evaluation, we generate 35 sequences, fixing $d_{min} = d_{max} = 2$, $l_{min} = 1$, and $l_{max} = 10$, where T is not fixed *a priori*. Using the generated data, we compare the training time and recognition time while changing the number of training data. The training time results are presented in Figure 4.7. The x-axis shows the number of training data. The y-axis shows the execution time for training. The upper, middle, and bottom lines respectively present the results of IS-HSMM, ILP-HSMM, and HSMM. Results show that three graphs are mostly increasing parallel, which shows that the difference between the

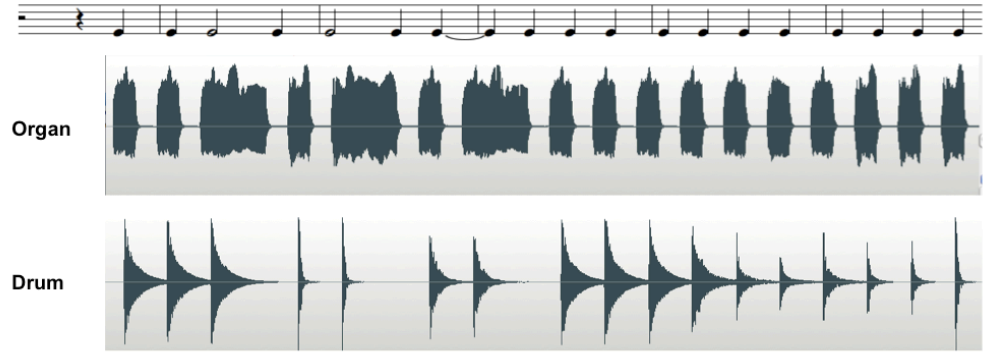


Figure 3.12: Music sound data.

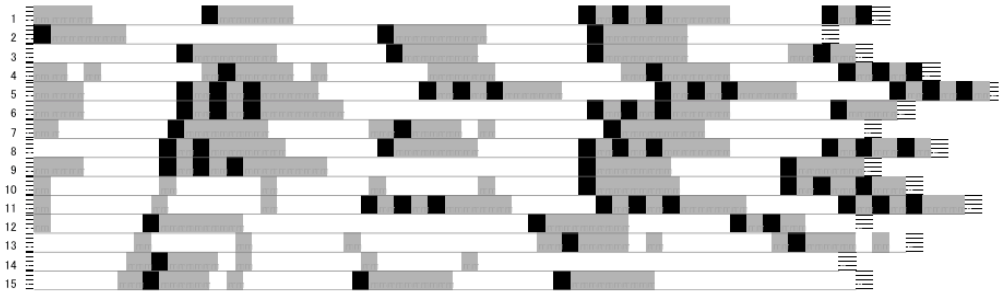


Figure 3.13: Example sequences generated using music sound data.

results of HSMM and IS-HSMM, and the difference between the results of HSMM and ILP-HSMM are both of a certain degree. Therefore, the training time of IS-HSMM and ILP-HSMM requires additional time, but the amount of the additional time does not increase exponentially.

Similarly, the execution time for recognition is shown in Figure 4.8. The x-axis shows the number of test data. The y-axis shows the execution time for recognition. The upper, middle, and bottom lines respectively present the results of IS-HSMM, ILP-HSMM, and HSMM. Results show that the amount of the additional time for recognition does not increase exponentially to the same degree as training. Stated differently, both the evaluation results of training time and recognition time reveal that it causes no severe difficulty for the execution times.

3.4.3 Recognition performance evaluation

This section presents the evaluation results of recognition performance comparing IS-HSMM and ILP-HSMM with HSMM. The evaluation metric is the recognition accuracy based on the *f*-measure calculated using
$f - measure = (2 \cdot \text{recall} \cdot \text{precision})/(\text{recall} + \text{precision})$, where precision $= TP/PP$, and recall $= TP/AP$. Here, the Predicted Positive (PP) is the number of models with likelihood calculated using (2.21) is maximum in all models. True Positive (TP) is the number of collected models in PP. Actually Positive (AP) is the number of labeled models.

Results are presented in Figure 3.16 and Figure 3.17. The x-axis shows Precision, Recall, and *f*-measure. The y-axis shows the score. The left, middle, and right bars respectively present the

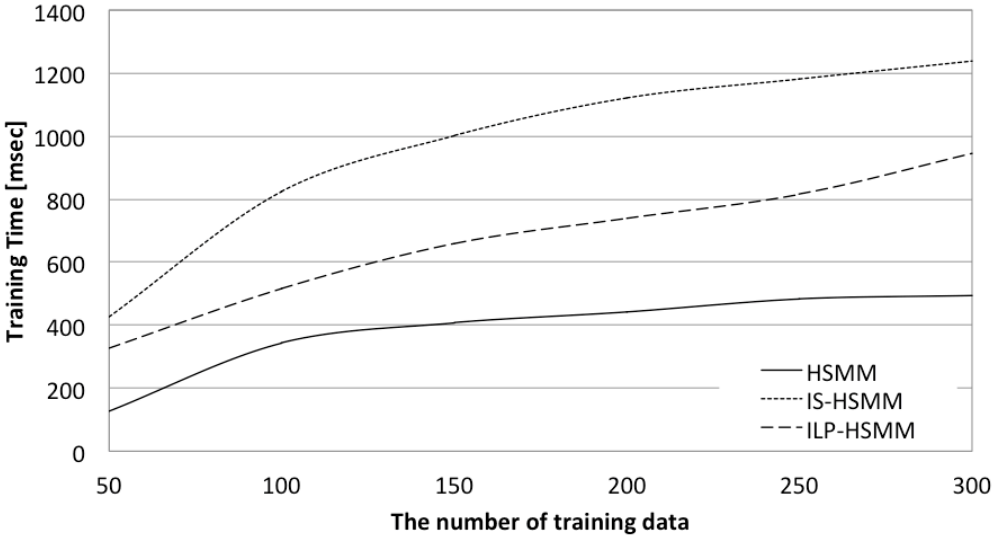


Figure 3.14: Execution time for training.

results of HSMM, IS-HSMM, and ILP-HSMM. Figure 3.16 shows the results obtained when the number of states is five, and Figure 3.17 presents the results obtained when the number of states is ten. Both results are the average scores of five repetitions. The results show that both the proposed models IS-HSMM and ILP-HSMM have higher recognition performance than HSMM. By comparing the results of IS-HSMM and ILP-HSMM, the scores of *f*-measure are similar, but the scores of recall and precision differ. IS-HSMM has a higher score for recall, but it has lower score for precision than ILP-HSMM. The next section presents detailed analysis of the performances of IS-HSMM and ILP-HSMM. Finally, comparison of the two results obtained when the numbers of states are five and ten shows that the recognition performance can be higher depending on the number of states increasing.

The earlier experiment includes observation symbols of only three kinds. To evaluate the performance of treating various durations and intervals with observation symbols of many kinds, we use the musical scale instead of the volume of the sound as observation symbols. Figure 3.18 shows the musical scale with stairs of example data. These are the some input data extracted from the evaluation data. The number on each graph signifies the label. Each value from 0.01 to 0.12 in 0.01 intervals is assigned to C, C#, D, D#, E, F, F#, G, G#, A, A#, B of the musical scale. If the volume is lower than a threshold, then the value of the sound scale label is zero. This is the *interval observation* in a sequence. The results of recognition performance using the data generated as described above are shown in Figure 3.19 and Figure 3.20. They present results of recognition performance evaluation when the numbers of states are 2 and 10. The scores are the average scores of five repetitions. Considering that it would be high performance when the number of states is greater than the number of observation symbols in HSMM, we assign 2 and 10 as the numbers of states in the experience to compare their performance.

When the number of states is 2, the recognition performance of HSMM is extremely low, but those of IS-HSMM and ILP-HSMM are much higher than HSMM. In addition, the results of IS-HSMM are much higher than ILP-HSMM. However, when the number of states is 10, the number of states is greater than the number of the observation symbols. At this time, the entire scores of HSMM, IS-HSMM, and ILP-HSMM are higher than 0.4. For the HSMM, the recall score gives the max score

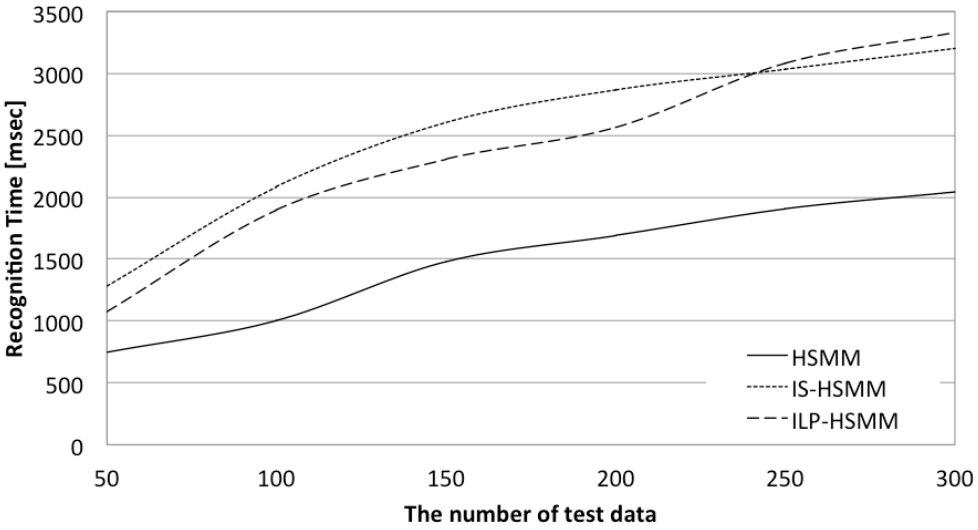


Figure 3.15: Execution time for recognition.

in all models but the precision score represents the lowest value. Therefore, the probability for each sequence using HSMM is similar to that of each other sequence. Then, whereas the average scores of precision, recall, and *f*-measure are more than 0.8 in IS-HSMM, the average score is about 0.7 in ILP-HSMM. As a result, when the number of states increases, the scores of IS-HSMM are higher than those of ILP-HSMM because increasing the states contributes to treatment of the transition probability from a state to another state. Therefore, IS-HSMM is effective for treating the order of the sequence precisely because the "interval" is represented with one of states and HMM can model the transition probability between two states.

However, regarding the input data shown in Figure 3.18 in detail, No. 4 input data are similar to No. 7; the No. 2 input data are similar to No. 10. It is difficult to distinguish the small time difference between two sequences with both IS-HSMM and ILP-HSMM even if the number of states increases. This difficulty might cause a decline of recognition performance.

Moreover, ILP-HSMM treats the state interval using the new additional parameter between two stationary states. If the state interval is mostly similar between static two states, then ILP-HSMM can model the length of the interval precisely, but it is difficult to model a sequence including various lengths of durations and intervals. Therefore, to treat sequential data of various kinds with durations and intervals, IS-HSMM would engender higher performance than ILP-HSMM. The following section presents evaluation results of modeling performance and analysis between ILP-HSMM and IS-HSMM.

3.4.4 Reproducibility performance evaluations between IS-HSMM and ILP-HSMM

This section presents the evaluation results of modeling performance, particularly addressing the performance of reproducibility. We calculate the performance of reproducibility and compare both IS-HSMM and ILP-HSMM. The performance of reproducibility signifies how precisely the model

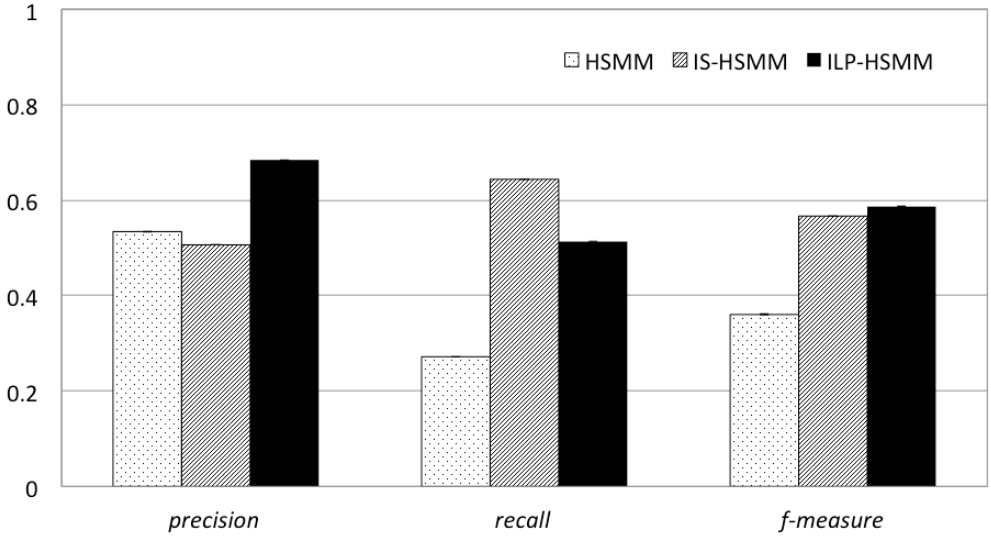


Figure 3.16: Recognition performance: the number of states is 5.

generates the original sequence, which is represented as r. The r is calculated as

$$r = \frac{\sum_{t=1}^{T}(w_t = o_t)}{T},$$

where $\boldsymbol{o}_{1:T}$ stands for the original sequence, T represents the time length of the original sequence, and $w_{1:T}$ denotes the generated sequence using the model parameter θ which is calculated using the original sequence. To calculate the equation presented above, we give the sequence length T and generate a sequence which has high likelihood using the forward algorithm with the set of parameters Λ. The generated sequence is the estimated sequence. Therefore, the performance of reproducibility indicates how precisely the model, i.e., the set of parameters Λ decided by the training phase, generates the original sequence.

First, we evaluate the performance of reproducibility when the number of states changes. Figure 3.21 presents the results of evaluating reproducibility using HSMM, IS-HSMM, and ILP-HSMM. The x-axis shows the number of states. The y-axis shows the performance of reproducibility. The number of observed symbols in sequence N is $N = 7$.

Results show that all models obtain higher performance of reproducibility when the number of states increases. The performance results of IS-HSMM and HSMM is mostly the same and IS-HSMM has a bit higher performance than that of HSMM. The results of ILP-HSMM show less performance when the states are fewer than six. They show higher performance when the number of states is greater than six. It represents that the number of states is more than the number of observed symbols; ILP-HSMM has higher performance of reproducibility than other models.

Then, we evaluate the performance of reproducibility when the number of intervals in a sequence changes. Figure 3.22 also shows the scores of performance of reproducibility of HSMM, IS-HSMM and ILP-HSMM. The x-axis shows the number of intervals in a sequence. The y-axis shows the score of performance of reproducibility. The number of sorts observed in a sequence is $N = 6$. One of the sorts is an interval. Results show that the performance of reproducibility of both models; HSMM and IS-HSMM decrease as the number of intervals increases, but that of IS-HSMM is higher than that of HSMM. Then, the results of ILP-HSMM is the highest performance in all models. It can obtain the highest performance irrespective of the number of intervals. Therefore, IS-HSMM can model the

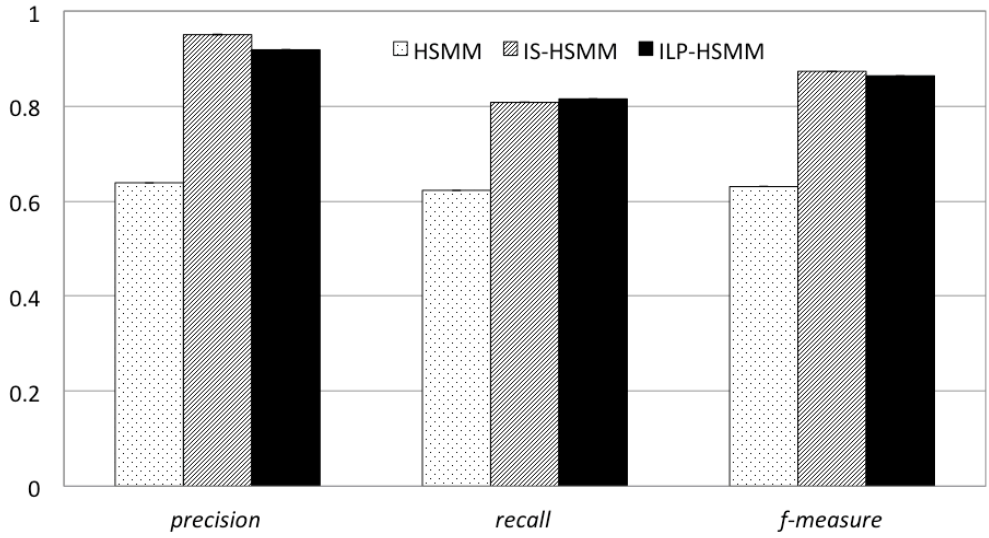


Figure 3.17: Recognition performance: the number of states is 10.

sequence with intervals more precisely than HSMM. The ILP-HSMM can model it most precisely of all models. Comparing two results of HSMM and IS-HSMM ensures that the proposed IS-HSMM can model the sequential data more precisely than HSMM by introducing the special state, i.e., the interval state and calculating the transition probability from the state before the interval state. In addition, the performance of IS-HSMM is much higher especially when the states are few and even if many intervals exist in a sequence. Comparing the other results for IS-HSMM and ILP-HSMM ensures that the proposed ILP-HSMM can model the sequential data more precisely than other models because it represents the length of intervals directly in the model.

As a result of the evaluation presented above, both the proposed extension models for HSMM have higher performance than HSMM, but ILP-HSMM can model the static interval between two states. However, it is more important for modeling the general duration and interval using a model with trained multiple data which have the same label. From the perspective of modeling generalization, the recognition performance of IS-HSMM has a higher score than other models, especially where the number of sorts of the observation symbols is larger. Therefore, we conclude that IS-HSMM has higher performance for modeling the general sequential data, not only for the data which have a static length of interval, but also for data which have various interval lengths.

3.5 Summary

The goal of this research was to model sequential data, including state duration and state interval, simultaneously. We specifically examined a hidden semi-Markov model (HSMM) to treat such sequential data, and proposed two extended models to treat a state interval in a sequence: IS-HSMM and ILP-HSMM. IS-HSMM introduces a special calculation technique to treat an interval state, where if the preceding state is an interval state, it models the transition from the second preceding state to the current state simultaneously. However, ILP-HSMM uses the Gaussian distribution as a length parameter, and trains with both preceding and subsequent states. Comparisons of recognition performance and elapsed time among IS-HSMM, ILP-HSMM, and HSMM show that both of the proposed models give higher performance than HSMM although they need additional calculation

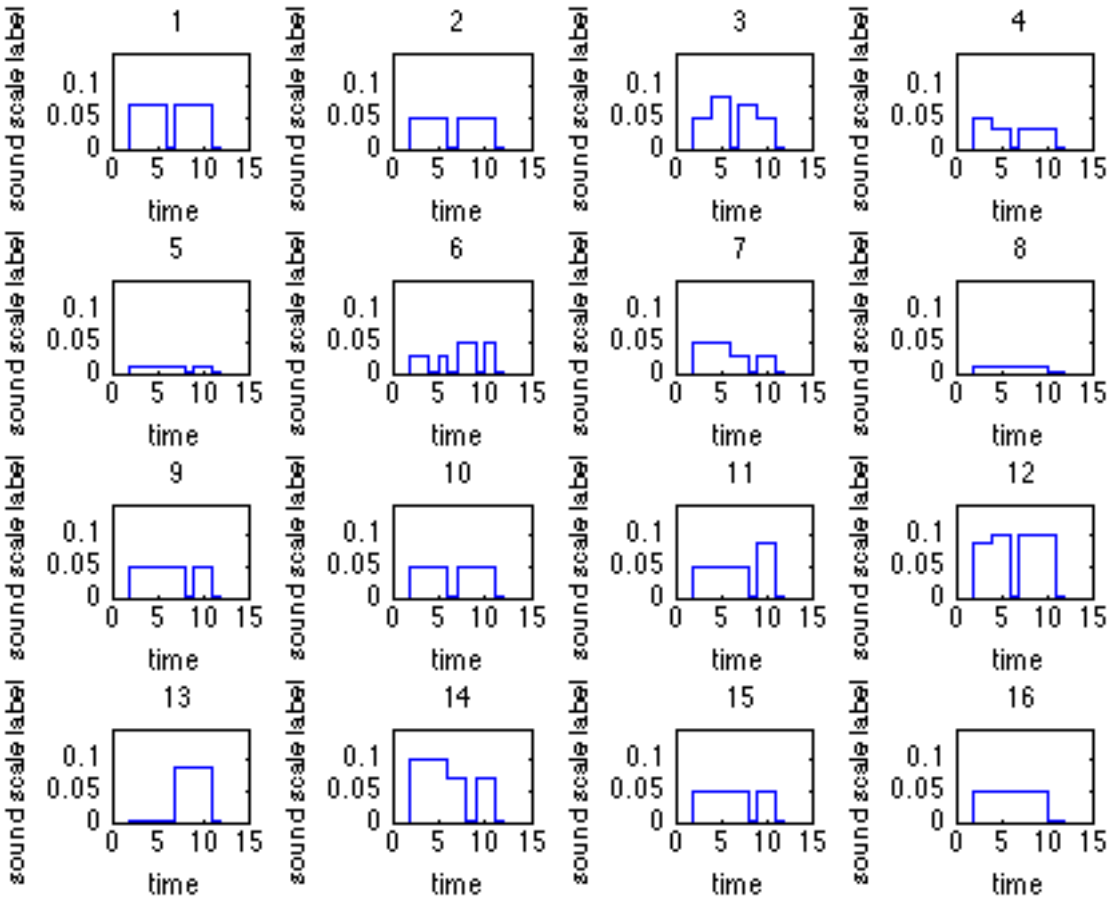


Figure 3.18: Musical scale of example input sequences and their labels.

costs. Comparison results between IS-HSMM and ILP-HSMM in terms of the modeling performance reveal that ILP-HSMM has higher performance than that of IS-HSMM.

As direction of future research, we intend to use our model to treat such actual sensing data which have a feature of rhythm or timing patterns. Although ILP-HSMM has higher performance in the evaluation, the concept of IS-HSMM is simpler than that of ILP-HSMM. Additionally, IS-HSMM can adopt another difficulty of analyzing sequential data, except for only treating intervals between states. In case the same state occurs frequently in a sequence, it is difficult to model the original sequence precisely without an interval. Therefore, we must evaluate the effectiveness of treating the original sequence using other application data, and finally extend the model further.

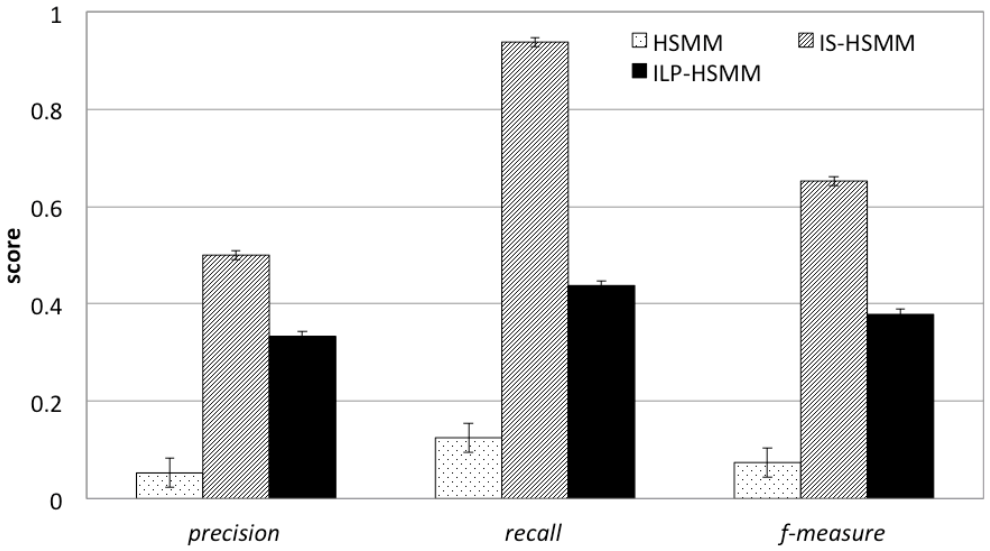


Figure 3.19: Recognition performance with music scale label: the number of states is 2.

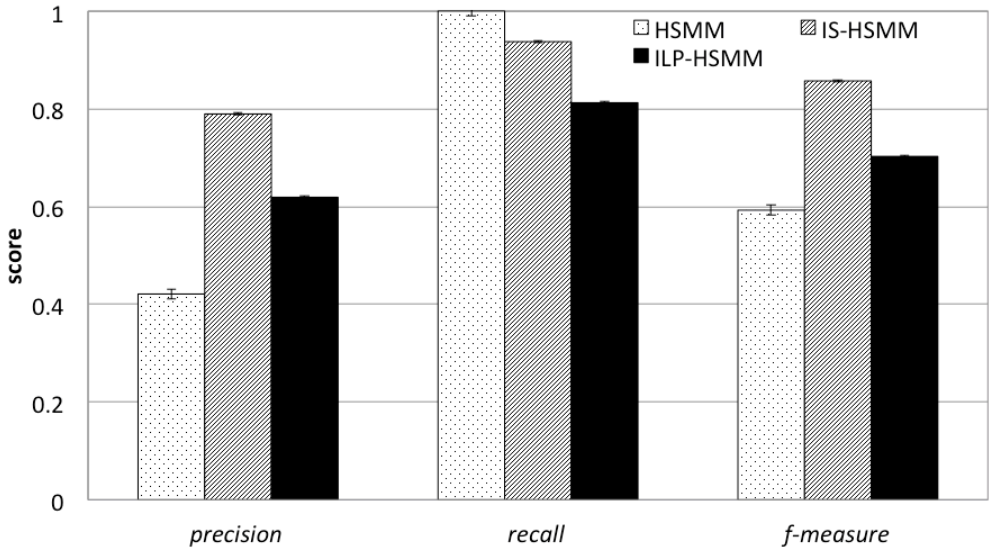


Figure 3.20: Recognition performance with music scale label: the number of states is 10.

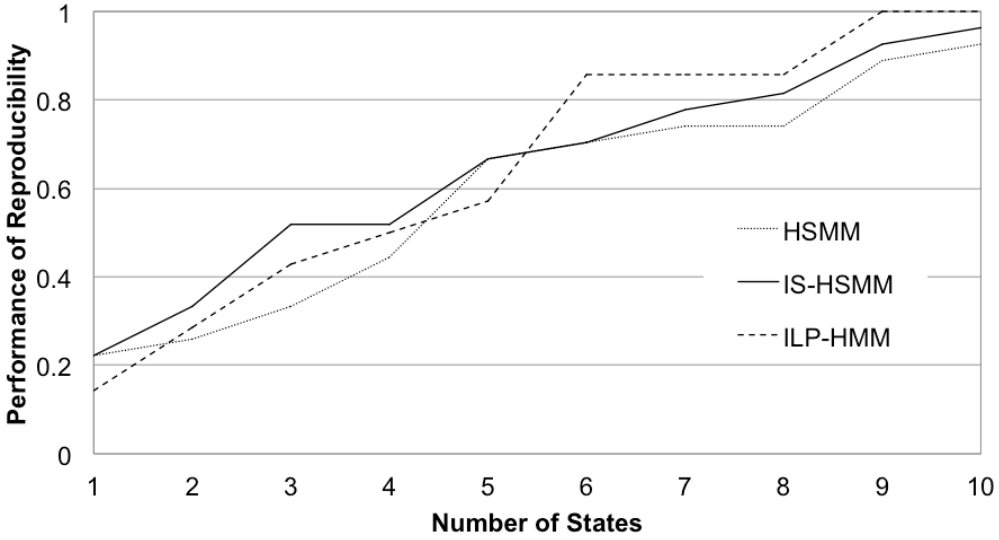


Figure 3.21: Modeling performance when the number of intervals increases.

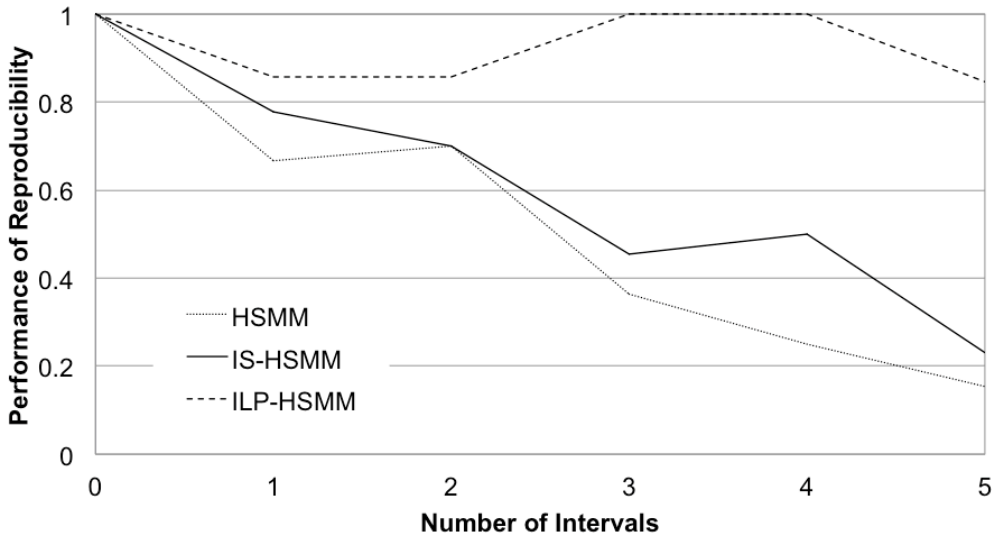


Figure 3.22: Modeling performance when the intervals become more numerous.

Chapter 4

Overlapped state hidden semi-Markov models

4.1 HSMM with multiple sequence input

Considering to dealing with multiple observation sequences, there are some extended HSMMs but it cannot support our requirement. Some extended models of HSMM deal with multiple sequence input [74, 76, 75, 52], and they assume that multiple observations are observed all the time simultaneously and an observation sequence is related to the other sequences. However, our target is abstracted event sequence and these sequence are grouped by choice. Therefore, the assumptions for the extended HSMM are different from our target in terms that the grouped sequences of our target are not related one another. Consequently, this section considers how to deal with the four requirements described in Section 2 based on the HSMM, and proposes a novel extended model, designated as overlapped state hidden semi-Markov model: OS-HSMM.

Considering the model of the grouped multiple sequences using HSMM, we first examine two straightforward extended approaches by addressing the fact that HSMM handles only single sequences. As explained later, both approaches fundamentally have technical problems. First, we points out the problems using Figure 4.1, where two pairs of two group sequences, i.e., Sequence 1 and 2, and Sequence 3 and 4, are compared. Whereas the lefthand side of Figure 4.1 presents the input sequences, the righthand side portrays the state sequences estimated by HSMM.

The first approach (Approach 1) models two sequences independently using HSMM. In this case, the estimated state sequences of the two pairs are identical, that is, the state sequences obtained from Sequence 1 and 3 are the same, and those from Sequence 2 and 4 are also the same. Because the difference between Sequence 1 and 3 is the continuous length of "a" and the average continuous length in each sequence is the same, this approach cannot discriminate the group of Sequence 1 and 2, and the other group of Sequence 3 and 4.

Meanwhile, the second approach (Approach 2) combines the two sequences into one single sequence at the same timeline, i.e., combining the two symbols locating at the same position in the vertical direction, which generates a new combined symbol. Subsequently, this model results in utilizing the combined single sequence using HSMM. When symbols are overlapped at the same time t, a new symbol is generated by combining the overlapped two symbols as "ac" in the figure. The advantage of this approach against the first one is that the estimated state sequence trained by these sequences are different each other as seen in the lower right illustration of Figure 4.1. However, because HSMM requires to assign the combinations of all symbols before the training, it is impractical to cover possible patterns of the combined symbols given any symbols. We face the limitation of

memory consumption and computation for the modeling.

Although each approach has the individual problem as seen above, the second approach is one possible way to model the group of sequence in terms of discrimination performance if the explosion of the number of combined symbols' patterns is solvable. Therefore, we tackle the problem of the combination explosion in Approach 2. More specifically, we consider the overlapped states occurring at the same time by introducing *negative state interval* of which *positive* value usually represents the time of the interval between two states that occur at different times.

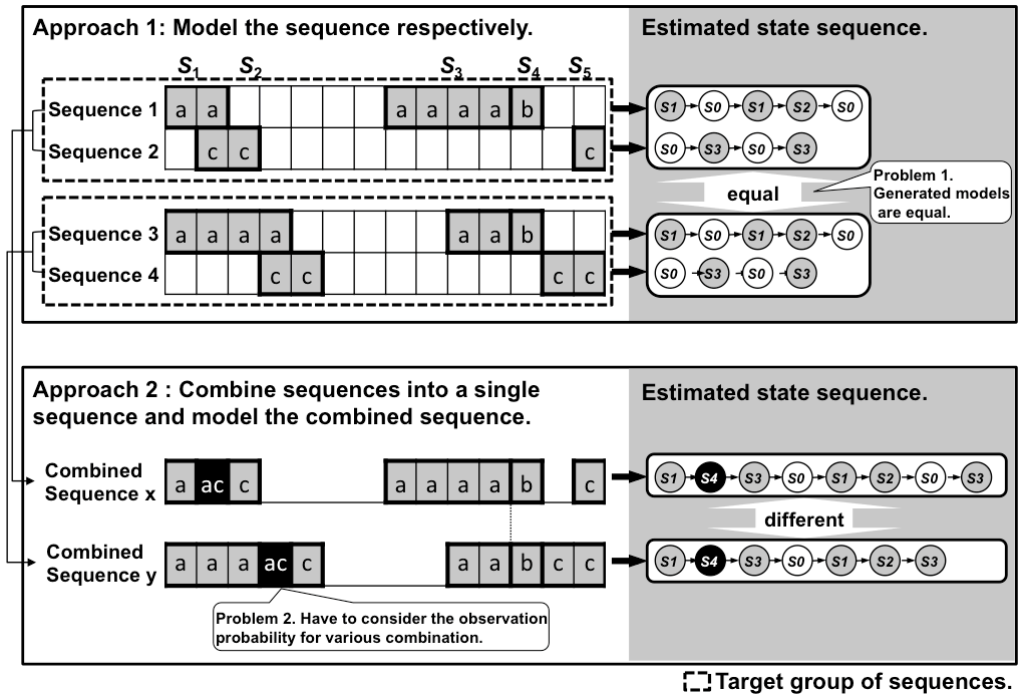


Figure 4.1: Problem of representative modeling.

4.2 Overlapped State HSMM (OS-HSMM)

This section proposes a novel model designated as overlapped state hidden semi-Markov model (OS-HSMM) to handle multiple sequences by translating it into one single sequence by extending HSMM. The fundamental concept is to use only the observed symbols as it is without combining and to model the overlap by newly introducing *overlap label* indicating the position of overlap and *overlap length probability* indicating the probability of the continuous time of the overlap label. The proposed method consists of three procedures; (a) sequence translation into a single sequence with the overlap label without using the combination of symbols, (b) training of the translated sequence with the overlap label by modeling continuous length of the overlap and interval, and (c) recognition and decoding.

4.2.1 Model training (inference) in OS-HSMM

The model training consists of two steps, where **Step 1** is to translate multiple sequences into a single sequence with the overlap label, and **Step 2** is to train the model using HSMM with the length of overlapped time.

Step.1 Sequence translation

This process is to translate multiple sequences into a single sequence with the overlap label. Figure 4.2 shows a conceptual example. Sequence 1 and 2 are the input sequences, and "Translated sequence" is an output sequence. When combining the input sequences in the vertical direction, an overlap of two states occurs at time $t = 4$ when the start point is set to $t = 1$. Here, we define the set of continuous observation, i.e., "aaaa" in Sequence 1 and "cc" in Sequence 2, as an *uniset* of observations. Then, we translate into a single sequence dealing with the uniset. Because the uniset of "cc" starts before the uniset of "aaaa" ends in this case, the uniset of "cc" is shifted to the positive direction as there exist no overlap. At the same time of shifting, the overlap label is assigned to the shifted uniset where the symbols occur at the same time in the original multiple sequences. The overlap label is shown as r in the figure.

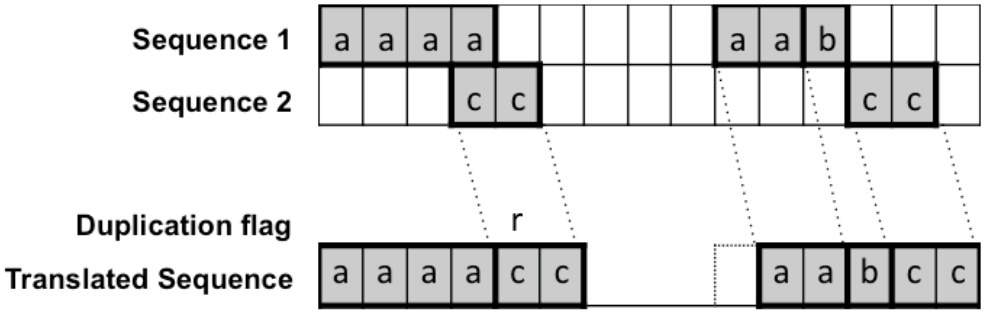


Figure 4.2: Modified sequence.

Step.2 Training with overlap label

To model the translated sequence with the overlap label, we introduce a new parameter which indicates the probability of time length when the symbols occur at the same time. It is noteworthy that this approach utilizes the overlap length probability to represent two types of length; state interval and state overlap in the same function.

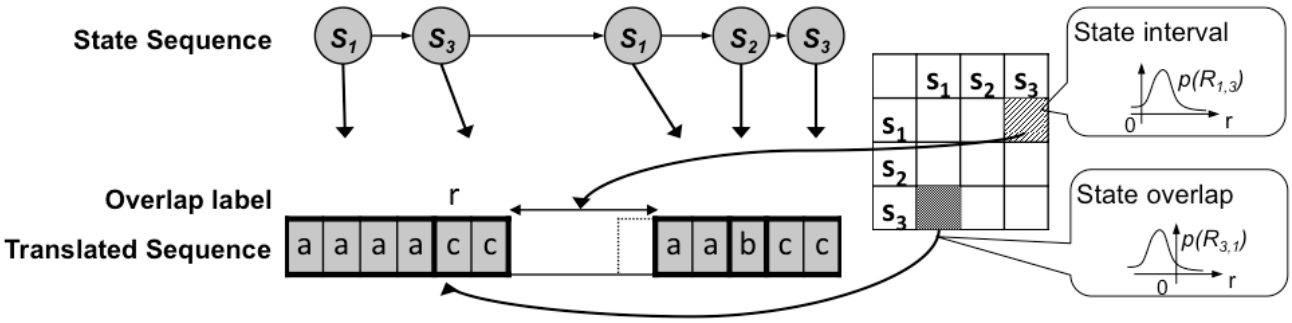


Figure 4.3: Proposed model.

Figure 4.3 shows the proposed concept of the new parameter $R_{i,j}$ representing the overlap length probability. $R_{i,j}$ is represented as the probability density distribution of the length. Then, $R_{i,j}$ is positive value when it represents the length of state interval. Meanwhile, allowing the negative value of $R_{i,j}$, we deal with the overlapped time, i.e., state overlap, by the negative value of $R_{i,j}$. We express $R_{i,j}$ by the Gaussian distribution using the continuous times of the overlap label r calculated by

$$p(R_{i,j}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(R_{i,j}-\mu)^2}{2\sigma^2}}, \tag{4.1}$$

where σ and μ present the variance and the mean of $R_{i,j}$, respectively. The observed length of interval and overlap is used as μ. If multiple different lengths are observed in the transition from state i to j, $R_{i,j}$ is generated for each μ, and the probability functions are integrated with multiple peaks as the total values of the probabilities is 1. By doing so, the proposed model can discriminate the state interval and state overlap by examining whether $R_{i,j}$ is positive or negative. Accordingly, the set of parameters used in OS-HSMM is defined as

$$\Lambda := \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}, \mathbf{D}, \mathbf{R}\},$$

where the elements of the parameter Λ take on $\mathbf{A}(i,j) = a_{(i,d_i)(j,d_j)}$, $\mathbf{B}(j,n) = b_{(j,d_j)}(\boldsymbol{o}_{1:d_j})$, and $\boldsymbol{\pi}(i) = \pi_{j,d_j}$, where $d_i \in D_{max}$ represents the duration of state i described in Section 2.3.2 and the proposed new parameter $p(R_{i,j} \in \mathbf{R} \in \mathbb{R}^{M\times M})$. The range of $R_{i,j}$ in (4.1) might influence memory consumption to generate the model. Therefore, the boundary of the probability value δ_r to ascertain the edge of the skirt of $p(R_{i,j})$ Therefore, we introduce the boundary of the probability value δ_r to ascertain the edge of the skirt of $p(R_{i,j})$. On generating the $p(R_{i,j})$, the calculation is terminated when the probability value becomes less than δ_r. The probability of $p(R_{i,j})$ is zero outside of the range of δ_r.

The transition probability and the emission probability are defined as the same as (2.3) and (2.4) in HSMM. These probability updates are calculated as

$$a'_{(i,d_i)(j,d_j)} = \frac{\sum_{t=1}^{T} \alpha_t(i,d_i) a_{(i,d_i)(j,d_j)} b_{i,d_i}(\boldsymbol{o}_{t+1:t+d_i}) \beta_{t+d_i}(j,d_j)}{\sum_{t=1}^{T} \alpha_t(i,d_i)\beta_t(i,d_i)}. \tag{4.2}$$

$$b'_{j,d_j}(\boldsymbol{o}_{t+1:t+d_j}) = \frac{\sum_{t=1}^{T} \delta(o_t, y_n)\alpha_t(j,d_j)\beta_t(j,d_j)}{\sum_{t=1}^{T} \alpha_t(j,d_j)\beta_t(j,d_j)}, \tag{4.3}$$

where the forward variable $\alpha_t(i,d_i)$ is defined as (2.5), and the following backward variable $\beta_t(j,d_j)$ is calculated as

$$\beta_t(j,d_j) = \sum_{i\in\{S\}\setminus\{j\}} \sum_{d_i \in D} a_{(j,d_j)(i,d_i)} b_{i,d_i}(\boldsymbol{o}_{t+1:t+d_i}) \beta_{t+d_i}(i,d_i). \tag{4.4}$$

g Here, $\delta(o_t, y_n)$ is defined as

$$\delta(o_t, y_n) = \begin{cases} 1 & \text{if } o_t = y_n \\ 0 & \text{otherwise.} \end{cases}$$

Finally, we update the set of parameters as $\Lambda = \hat{\Lambda}$ until the difference between the log-likelihood from the previous one converges, where the log-likelihood is calculated using (2.20) where $\alpha_T(j,d_j)$ is calculated by (2.5) when $t = T$ at the end of the sequence.
The trellis diagram for forward calculation is described in Figure 4.4.

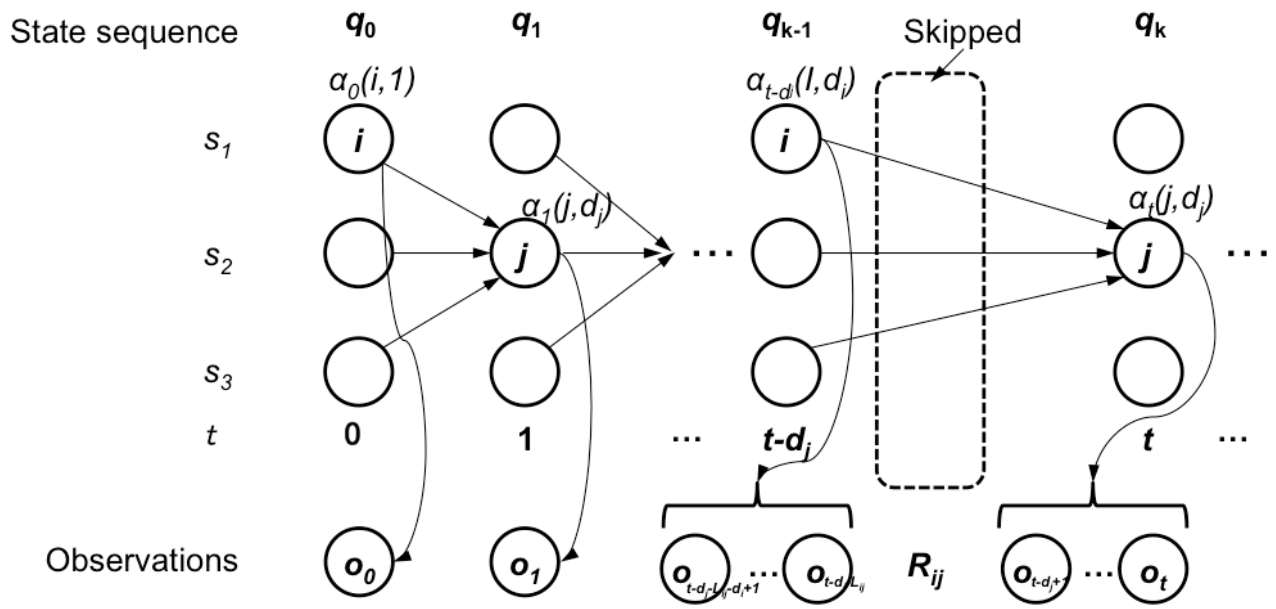


Figure 4.4: Trellis diagram for forward calculation of OS-HSMM.

4.2.2 Decoding and recognition using OS-HSMM

First, the grouped sequences are translated into a single sequence with the overlap label likewise as described in Section 4.2.1. The length of the original sequence T, and the overlap label sequence are stored together with the translated sequence. We first explain the decoding calculation, and then explain the recognition calculation.

Decoding

In the decoding phase, we generate a single sequence using the parameters without R as the same as the conventional HSMM while $t < T + \delta$ where δ is the enough length of overlap time for the sequence. If we generate the same sequence with the training sequence, it is set to $\delta = 0$. The probability of generating the single sequence is calculated using the forward algorithm used in HSMM. For each model, it recursively calculates the forward variable and the probability for each state using $P(\boldsymbol{o}_{1:T}) = \sum_{i=1}^{M} \alpha_T(i, d_i)$, which is the marginal probability distribution, where

$$\alpha_t(j, d_j) = \left[\sum_{i=1}^{M} \alpha_{t-1}(i, d_i) a_{(i,d_i)(j,d_j)} \right] b_{i,d_i}(\boldsymbol{o}_{t-d_i+1:t}). \tag{4.5}$$

Here, we denote explicitly the probability as $P(\boldsymbol{o}^*_{1:T}|\Lambda^z)$ using the parameter set of model z, i.e., Λ^z, where $z \in \{1, 2, \cdots, Z\}$ and Z is the total number of models. Finally, the label that has the maximum $P(\boldsymbol{o}_{1:T})$ for the observation sequence is selected as the recognition result. Thus, the model z^* that has the maximum probability $P(\boldsymbol{o}^*_{1:T}|\Lambda^z)$ among all Z models is selected as the result of the recognition. For each transition from state i to j, the overlap length probability matrix is examined by assigning i and j to $p(R_{i,j})$. Then, the maximum likelihood length calculated by

$$r_{i,j} \quad := \quad \arg\max p(R_{i,j}). \tag{4.6}$$

is chosen as the interval length of the overlap length from state i to j. When the $r_{i,j}$ is positive value, the interval symbols are inserted according to the absolute value of $r_{i,j}$ between state i and j. When the $r_{i,j}$ is negative value, the symbols from state i and j are overlapped according to the value. The

time length of overlap, $R_{i,j}$ that has the maximum probability where transits from state i to j, is continuously selected until the range satisfies $T < T'$.

Recognition

For recognition calculation, there are multiple target sequences. Before recognition, these sequences are translated into a single sequence with the overlap label described in Section 4.2.1. When translating multiple sequences into a single sequence, the original sequence length T, and the overlap label sequence are stored together. Where t which has r, the transition probability using α is calculated using the Viterbi algorithm for estimating the probability of a model [55] as the same as HSMM. The pair of the model with the overlap length probability and its label that is expected to be estimated are stored as candidates for estimation. The recognition label that denotes the estimated result is selected when the model has the maximum likelihood estimate by calculating the following equation by adding $p(R_{i,j})$ shown in (4.1) for (2.5) in each model.

$$\begin{aligned}\alpha_t(j, d_j) &:= \max_{s_{1:t-d_j}} P(s_{1:t-d_j}, s_{t-d_i+1:t} = j, \mathbf{o}_{1:d_j}|\Lambda) \\ &= \max_{i \in S\setminus\{j\}, d_i} \left\{\alpha_{t-d_j}(i, d_i) \cdot a_{(i,d_i)(j,d_j)} \cdot b_{i,d_i}(\mathbf{o}_{t-d_i+1:t}) \cdot p(R_{i,j})\right\}.\end{aligned} \tag{4.7}$$

$\alpha_t(j, d_j)$ at $t = T$ is calculated as the maximum likelihood estimate as (2.20).

The overlap length probability is calculated simultaneously with the calculation of the parameter of the likelihood using the transition probability recursively. Finally, the model which has the maximum likelihood estimate is selected as the recognition results. Pseudocode is shown in Algorithm 4.

4.3 Evaluation

To show the effectiveness of our proposed model, we evaluate recognition performance and modeling performance to verify that the proposed model OS-HSMM can discriminate the group of multiple sequences. We also measure total time of the training and the recognition to evaluate the calculation cost and finally, we evaluate our model using music sound data as a practical application data. Here, we use two types of HSMM. HSMM and HSMM (Oracle) represent Approach 1 and Approach 2 described in Section 4.1 respectively.

Experimental setup

We prepare two types of dataset; dataset-A and dataset-B shown in Figure 4.5. Dataset-A consists of data that have similar structure whereas the structure of the data of dataset-B is different. More specifically, such similar structural data are generated by changing only the length of the duration and interval with small kinds of symbols. The combination of (a_1,b_1), (a_1,b_2), $\cdots$ are dataset-A, and the combination of (c_1,c_2), (c_2,c_3), $\cdots$ are dataset-B. We use 50 for dataset-A and 46 for dataset-B by removing the exactly the same combination. The length of each sequence is $T = 22$ and there are from 3 to 5 state overlap in each combined sequences. The maximum iteration of the recursive forward/backward calculation is set to 30.

The recognition performance is calculated as

$$score = \frac{\text{Number of data that the recognition result is correct}}{\text{Number of dataset}}.$$

The modeling performance is calculated as

$$m = \frac{\sum_{t=1}^{T}(w_t = o_t)}{T},$$

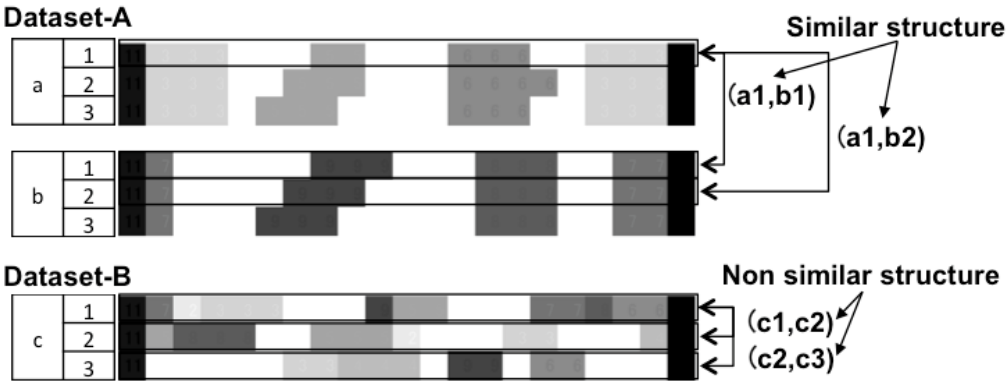


Figure 4.5: Evaluation data.

where w_t is the output symbol from the trained model at time t. In terms of modeling performance, we show the comparison between OS-HSMM and HSMM (Oracle) to compare with the oracle performance. The total time for the computation is also measured for each training and recognition by changing the number of dataset.

Results

Table 4.1 shows the results of the recognition performance. The results reveal that OS-HSMM provides higher performance when the data include the similar structural data even though both of the performances are competitive for the non-similar structural data.

Table 4.1: Recognition Performance.

	HSMM	OS-HSMM
Dataset-A (similar structure)	0.52	**0.72**
Dataset-B (non-similar structure)	**0.96**	0.93

Figure 4.6 shows the results of modeling performance, where x-axis is the data label, and y-axis is the modeling performance. It is calculated as

$$m = \frac{\sum_{t=1}^{T}(w_t = o_t)}{T}.$$

For all the datasets, the modeling performance of OS-HSMM is mostly the same as that of HSMM. By analyzing the results, we also find that HSMM (Oracle) has higher performance when the data include various symbols as non-similar structural data, but OS-HSMM has higher performance for similar structural data. Therefore, OS-HSMM is effective for discriminating the similar group of sequences.

Next, we evaluate the elapsed time of training and recognition. Figures 4.7 and 4.8 show the results, where x-axis and y-axis represents the number of dataset and the total time, respectively. As the number of datasets increases, the difference between HSMM and OS-HSMM increases accordingly. Because OS-HSMM calculates a sequence with length probability parameter whereas HSMM calculates each sequence independently, the calculation cost of OS-HSMM is lower than that of HSMM. From the evaluation results, we can conclude that OS-HSMM reduces the calculation cost while keeping the modeling performance high.

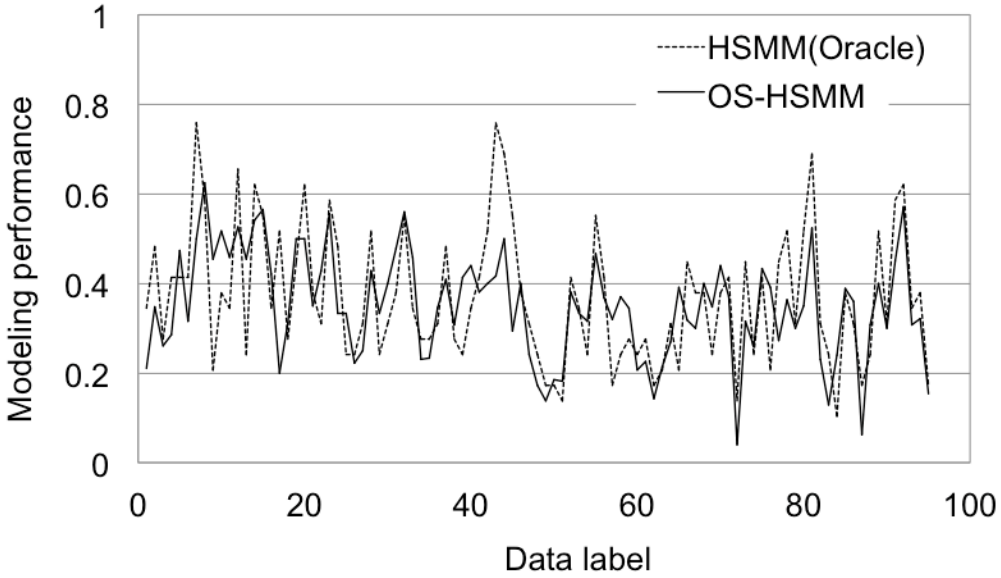


Figure 4.6: Modeling performance.

Results using music sound data

To evaluate the proposed model in real-world application data, we use music sound data that are played by 6 different instruments, a grand piano, horn, flute, pipe organ, acoustic guitar, and drums. The music data are divided into bars, and the length of the sequence is $T = 87$. Example data are shown in Figure 4.9. Sequence (b) is generated by shifting the sequence (a) to the positive direction without changing the window position of bars. The shifting length is $T + \alpha$, and α is set to 30. The set of two sequence is the input data; a_{n+1} and b_n. The classification of sound-pressure is used as observed symbols instead of sound pitch. The kinds of symbols are *high*, *low* and *interval*. The number of labels is 156.

We evaluate the recognition performance using the data in comparison with OS-HSMM, HSMM, and HSMM (Oracle). Table 4.2 shows the results. OS-HSMM has the highest performance in all. While the conventional methods cannot handle the length of overlap and interval properly, OS-HSMM deals with it efficiently.

Table 4.2: Recognition performance using music sound data.

	HSMM	HSMM (Oracle)	OS-HSMM
Accuracy	0.42	0.48	**0.61**

Results using weather data

Then, we use weather data as another practical application data because it is easy to obtain from the website of Japan Meteorological Agency [1]. We extract the sequence of weather overview separated by month from Jan. 2013 to Aug. 2017. Different labels are attached for each day weather overview sequences. We convert the weather overview for four symbols as the actual weather overview information is in detail because it can be easy to discriminate the difference with the model in case that the symbol is too detail. The four symbols are *sun*, *rainy*, *cloud*, and *other* as interval. Figure 4.10

[1] http://www.jma.go.jp/jma/indexe.html

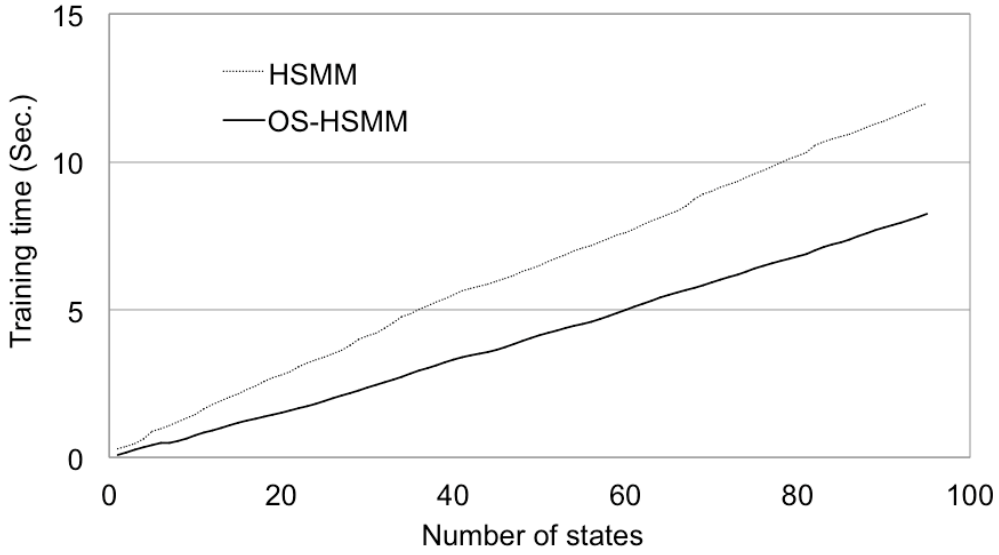


Figure 4.7: Elapsed time of training.

shows the example sequences of weather data. Likewise the previous experiment, we combine n-th sequence with $n+1$-th sequence as the group of sequences. The number of labels are 55, and we calculate the likelihood to generate the observation sequences by using the estimated parameters trained by each 55 group of sequences. Figure 4.11 shows the results when calculate the probability likelihood for label 1 sequence with all 55 models. It can obtain the highest score for the correct label label 1, the scores of label 29 and label 37 are close. Therefore, we extract each sequences of label 1, 29, 37 and 2, 30, 38 for analysis.

Figure 4.12 and Figure 4.13 show the individual sequence of label 1, 29, 37, and label 2, 30, 38. In each figure, there are three sequences and each line shows the sequence and each symbol shown in different colors. From the view point of individual sequences, it seems it does not similar. However, the sequences combined label {1, 2} and {29, 30} and {37,38} seems it is similar. Figure 4.14 shows the combined sequences. Therefore, it is considered that OS-HSMM is suitable for the group of sequences that each sequence has different observation symbols, but it needs additional challenge to deal the group of sequences as obtained by the same type of sensors.

4.4 Summary

Conventional approaches for multiple observation data sequence modeling are hierarchical models. However, they assume the relationship between the observation sequence and the state sequence, or introducing some constraints to the states emitting observations. However, from the view point of sensing or monitoring data analysis, the target events do not continuously occur, rather occur intermittently, and some of the sequences have no characteristic feature in the event occurrence pattern. If both of the event occurrence patterns in the multiple sequences are observed frequently, it is difficult to train as a different model. In case of combining multiple sequences as a single sequence before training, the combination explosion problem arises in the practical calculation. Therefore, we translated the multiple sequences into a single sequence leaving the information of the overlap, and propose a new model to treat the multiple sequences with the overlap label by translating multiple sequence into a single sequence. We have presented a new expanded HSMM, dubbed Overlapped

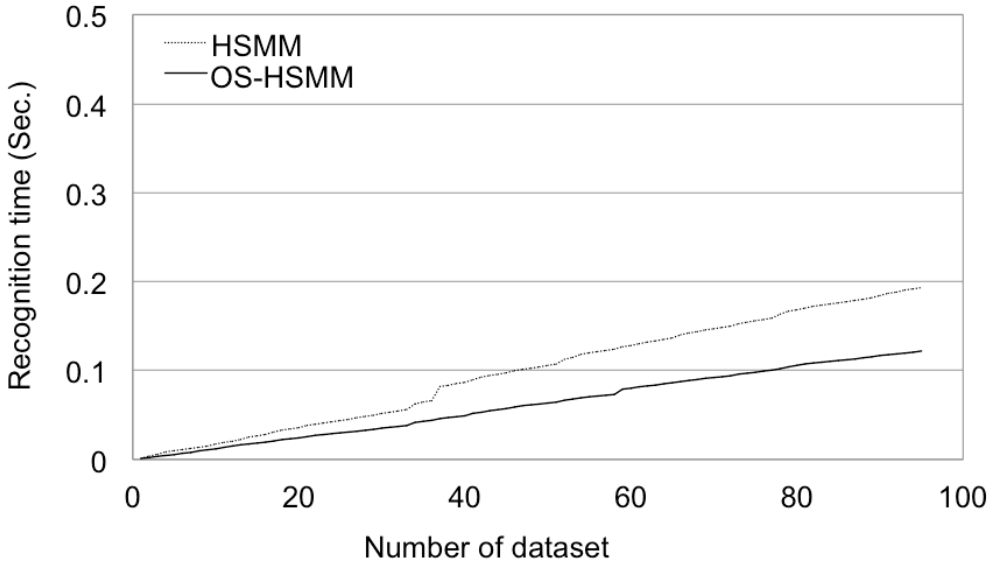


Figure 4.8: Elapsed time of recognition.

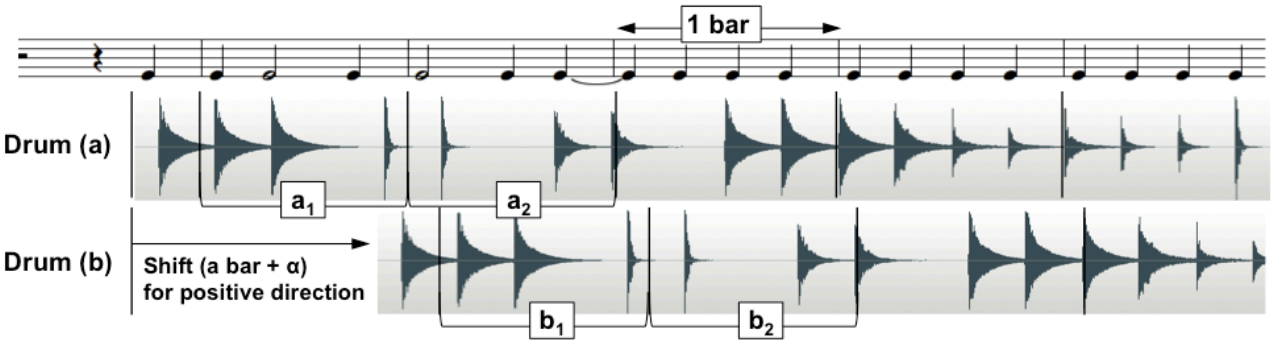


Figure 4.9: Music data.

State Hidden Semi-Markov Model (OS-HSMM). The evaluation results present the effectiveness of the modeling performance, recognition performance, and the calculation time.

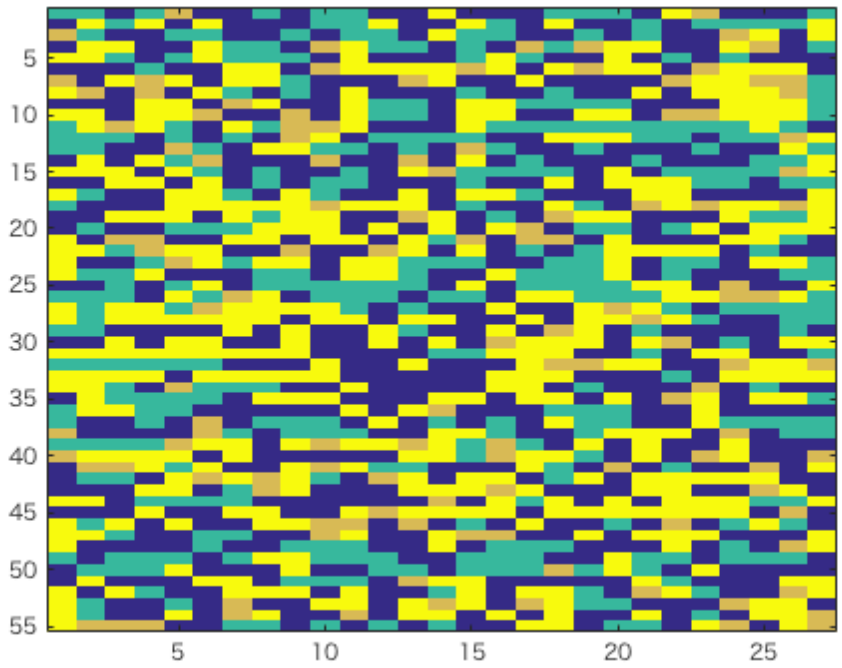


Figure 4.10: Sample data of weather data.

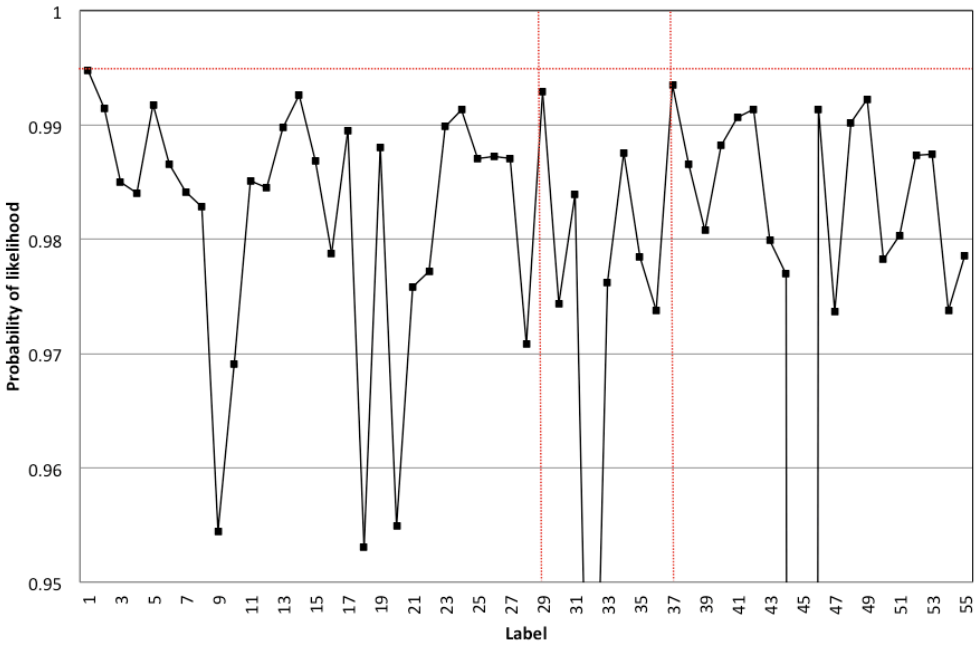


Figure 4.11: Results of likelihood score for each labels (Label 1 is correct).

Algorithm 4 Algorithm for training and recognition in OS-HSMM.

Require: Input
Training sequences 1 : $\boldsymbol{o}^{z1}_{1:T_r} = \{o^{z1}_1, \cdots, o^{z1}_{T_r}\}$,
Training sequences 2 : $\boldsymbol{o}^{z2}_{1:T_r} = \{o^{z2}_1, \cdots, o^{z2}_{T_r}\}$,
Testing sequences : $\boldsymbol{o}^{*}_{1:T_t} = \{o^{*}_1, \cdots, o^{*}_{T_t}\}$.

Ensure: Training phase

(Translated sequence, overlap label with t) $\leftarrow$ Proceed step 1 (Sequence Translation)
for $z = 1$ to Z **do**
  Assign random values to the OS-HSMM parameters $\Lambda^z = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}, \mathbf{D}, \mathbf{R}\}$, D_{max}, and $\alpha_{t(j,d_j)}$ and $\beta_{t(j,d_j)}$.
  for $h = 1$ to H **do**
    for $t = 1$ to T_r **do**
      Calculate $\alpha_{t(j,d_j)}$ and $\beta_{t(j,d_j)}$ using (2.5) and (4.4).
      if $\theta_h - \theta_{h-1} < \epsilon$ **then**
        Calculate $p(R_{i,j})$ from the continuous number of overlap label using (4.1).
      end if
      Update parameters Λ^z using (4.2) and (4.3).
    end for
    Calculate θ_h using (2.5) and (2.20).
    if $\theta_h - \theta_{h-1} < \epsilon$ **then**
      break
    end if
  end for
end for

Ensure: Recognition phase

for $z = 1$ to Z **do**
  for $t = 1$ to T_t **do**
    Prepare Λ^z from the results obtained in the training phase.
    Calculate $\alpha_t(j, d_j)$ using (4.7).
  end for
  Calculate $P(o_{1:T_t}|\Lambda^z)$ using $\alpha_t(j, d_j)$.
end for
Select the model z^* that has the maximum value for $P(o^*_{1:T_t}|\Lambda^z)$.
Return Model z^* and its probability $P(\boldsymbol{o}^*_{1:T_t}|\Lambda^{z^*})$.

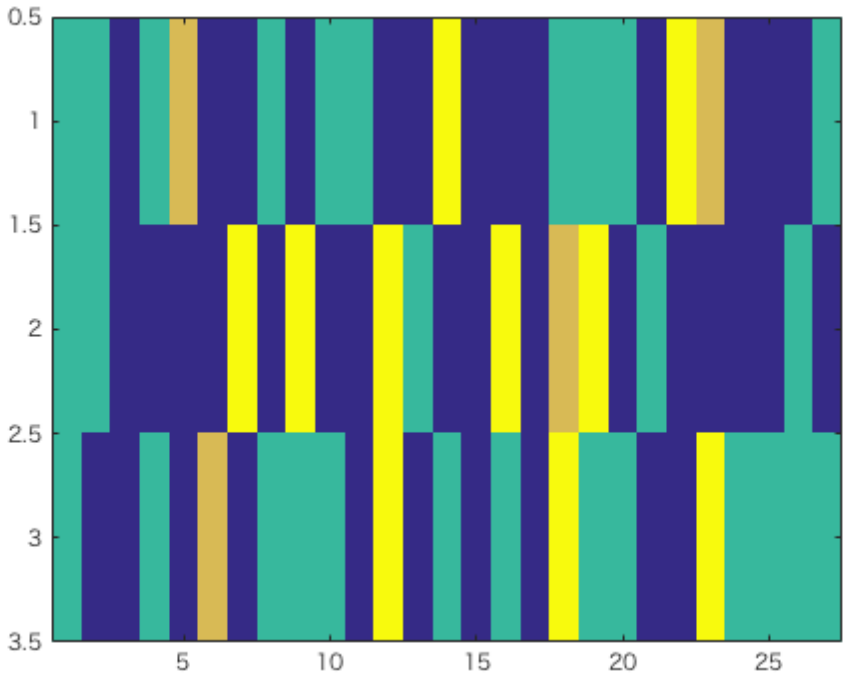


Figure 4.12: Sequence of label 1 and 29 and 37.

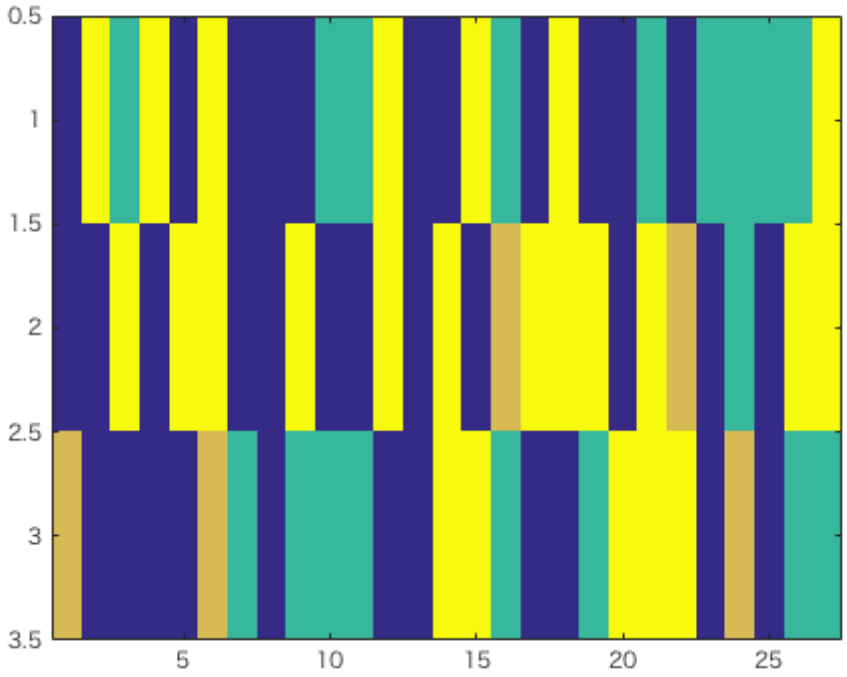


Figure 4.13: Sequence of label 2 and 30 and 38.

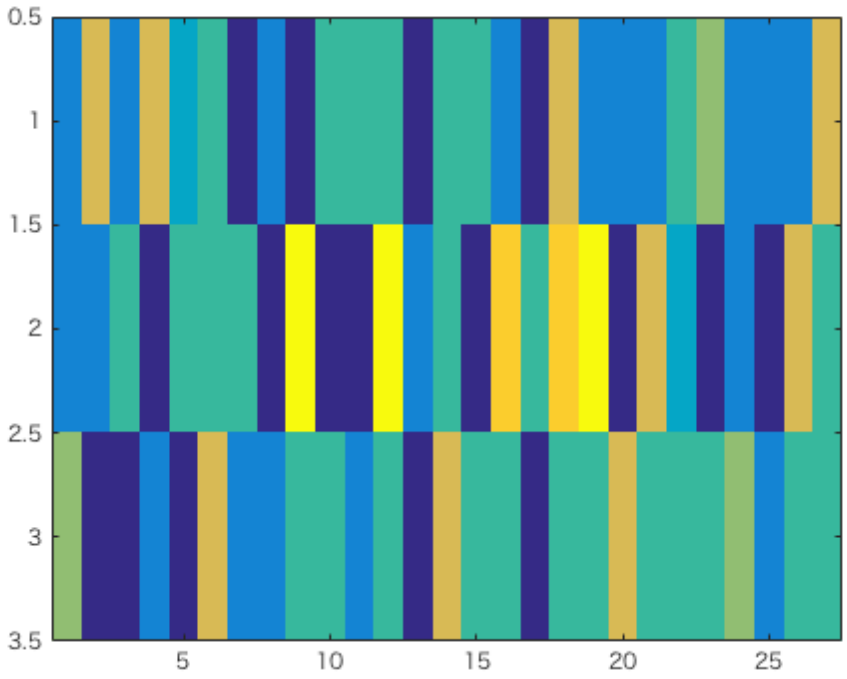


Figure 4.14: Combined sequence of label {1, 2} and {29, 30} and {37,38}.

Chapter 5

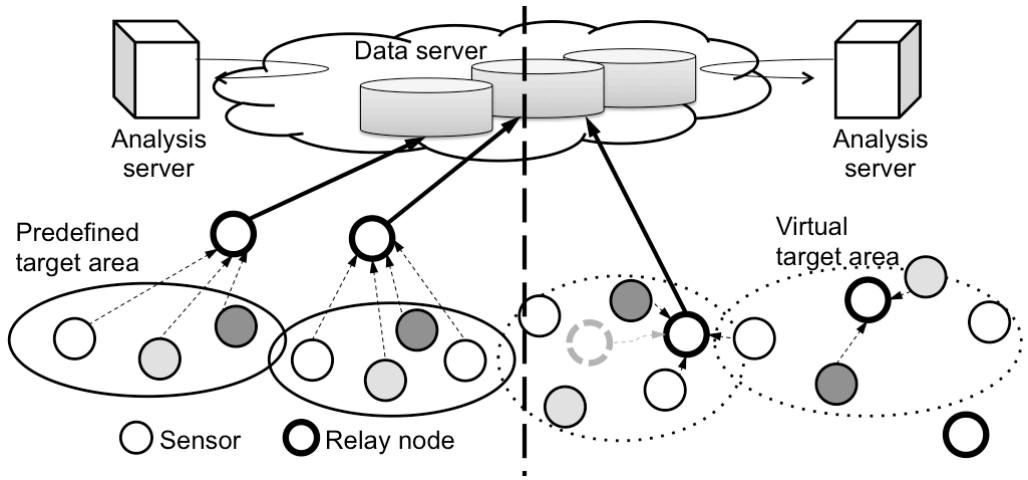
Grouped sequential data aggregation

5.1 Studies for data aggregation and management

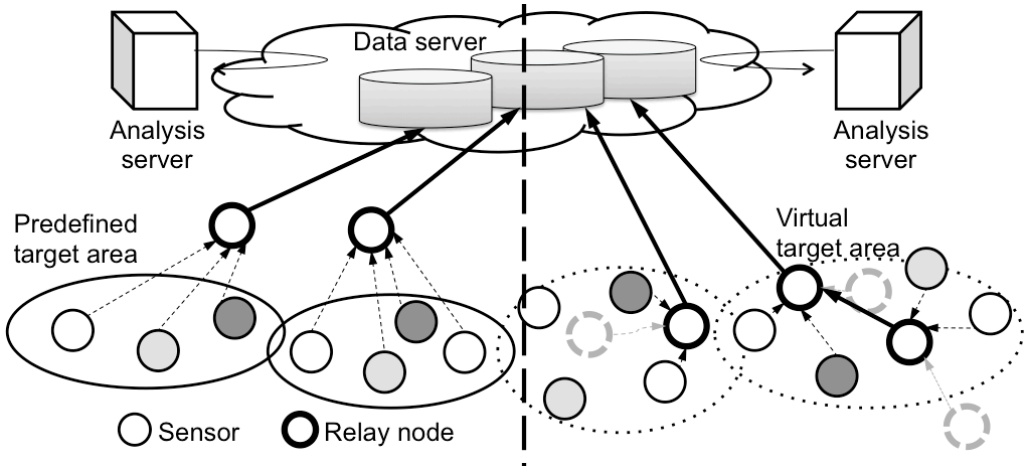
5.1.1 Motivation

This section presents the goal of the sequential data aggregation and describes the related work and the difference between them. Regarding the aggregation of the grouped sequential data, there are two problems. From the view point of easily aggregation by scattering sensors or devices around the area, we need to prepare the terminals which play roles of access points. It means that it need additional high cost or labor to prepare and establish the access points to aggregate the data from these sensors In spite of price reduction of sensors. On the other hand from the viewpoint of generating freely group, there is a constraint in the condition that the predefined target area consists of a prepared target access point. Figure 5.1 shows the conventional data aggregation methods and the goal of our data aggregation comparing from the viewpoint of the target area. The left side of the figures shows the conventional situation of predefined target area with prepared access points and the right side of them shows the concept situation of virtual target area which can be freely changed. Figure 5.1(b) shows the situation after a certain period since Figure 5.1(a) is observed. The conventional data aggregation is general and the prepared access points can stably aggregate the sensing data in the predefined target area. However, once the constitution of devices or the grouped sensors is expected to be changed, it needs to redefine the target constitution and reestablish the network configuration including the condition of device allocation. There is large barrier to realize the freely grouping. Therefore, we take an approach to realize the freely grouping access aggregation with the virtually defined target area with the devices or mobile terminals existing the target area. Figure 5.2 shows the concept of how to aggregate sequential data by the mobile terminals existing in the virtual target area in detail. The data in the virtual target area are aggregated and relayed by the existing devices in the area and finally send to the data server via network. The requirement to achieve situation with virtual target area via wireless terminals without stable and prepared access points, it is assumed that the size of relayed data is smaller than several MB because the sensing data is assumed to be poor, the requirement distance between two relay terminals are dozens meters as Bluetooth or Wi-Fi ad hoc communication, and the expected transmission rate is from 125kbps to dozens Mbps. It is considered that the time of communication can be more than 10 seconds when the terminals are moved with walking person. However, it is considered that the time of communication is less than several seconds when the terminals are moved together with runner or automatic vehicles. Therefore, it is assumed that the size of datum can be transmit with used communication type within a short time. The rest of this section shows the conventional data aggregation and the difference between

these methods with our approach.



(a) State of a change with time ($t = t_1$).



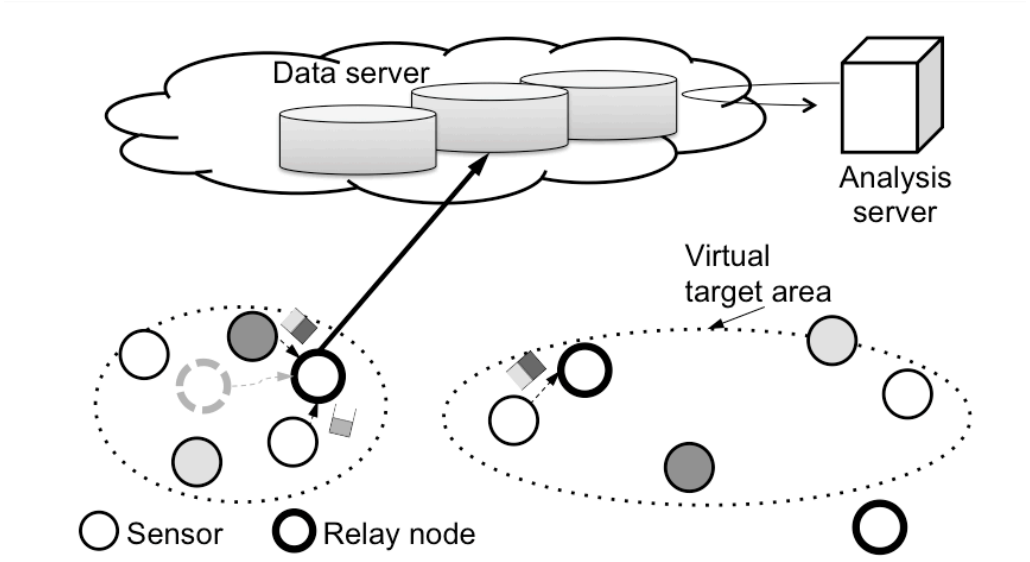
(b) State of a change with time ($t = t_2$).

Figure 5.1: Predefined target area vs. virtual target area $t_1 < t_2$.

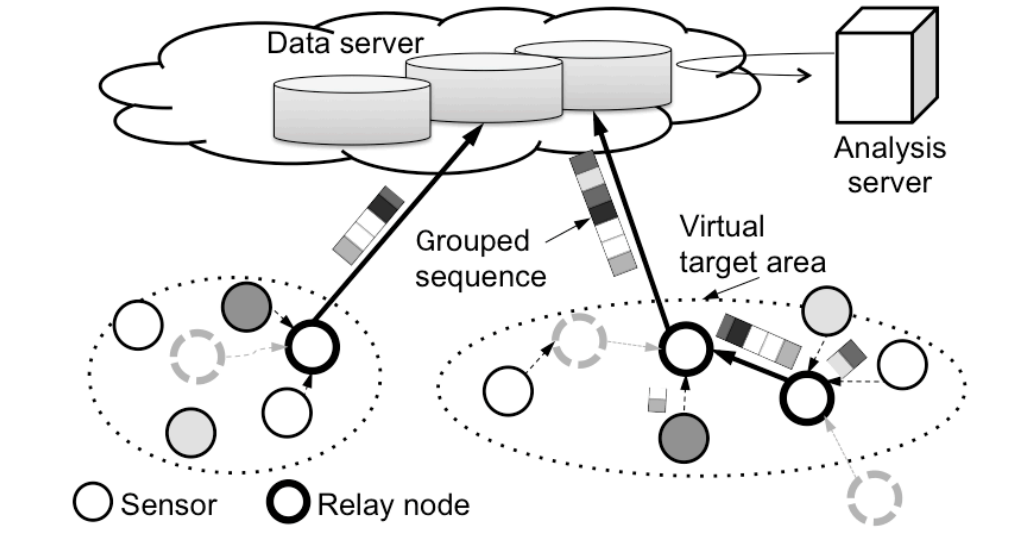
With the advance of mobile devices equipped with a short-range wireless communication technology like IEEE802.11 and Bluetooth, new communication services inside physically local area are growing rapidly [81]. The current advanced communication technologies enable passing strange people on a street to communicate each other, and exchange information during a very short time period. This application, often called an opportunistic application, might be, for instance, a real-time short message exchange application, and an individual profile exchange application. A real-time battle game on a portable game device is also an example of rapidly growing applications [82, 83, 84]. This successfully achieved explosive popularity, and produced a big business market. However, these application services are effective and enjoyable only for people at the same area at the same time since the communication occurs in a real-time manner. If a communication with time-offset inside a designated area is achievable, it would create a new communication style promoting strong emergence of a huge number of completely new applications. Therefore, a new storage capability and mechanism to keep information in a certain area is expected to enhance short-range communication.

Considering store data itself, most data in this type of communication may decrease in the value as time passes and finally can be discard since its meaningfulness and usefulness are temporary only

in a limited physical. Remote access to this data is also not needed. From these viewpoints, Cloud-based storage is too high functional and expensive since the network infrastructure and the network interface module on every device are required.



(a) Aggregation from sensors at time $t = t_1$.



(b) Grouping and uploading the aggregated data to the server at time $t = t_2$.

Figure 5.2: Concept of virtual target area and sequential data aggregation where $t_1 < t_2$.

In this section, we propose a new area-based distributed mobile storage without an infrastructure network. This is achieved by relaying and sharing information across multiple mobile terminals via a short-range wireless communication. This brings location-related "virtual storage" by collaborative terminals without network infrastructure. Practically, when mobile devices completely disappear from the target storage area, the store data cannot be kept any more. It is, however, acceptable for the viewpoint of the opportunistic communication applications mentioned above, and this platform aims at not a persistent storage, but a temporally storage. Hereafter, we call the data to be stored

as store data, and call the area where the store data is kept as storage area.

The rest of this section summarizes related works. Section 5.2 shows our proposed area-based collaborative distributed mobile storage. Proposed platform architecture and its system behavior are described in detail. Section 5.3 details a relay control algorithm. The algorithm provides relay timing and terminal determination algorithm and relay area control algorithm. Section 5.4 evaluates our proposed relay data control algorithm. Finally, we mention concluding remarks and insights for future works. In the rest of this chapter, IEEE802.11 is assumed as the short-range wireless communication technology.

5.1.2 Related work

A mobile ad hoc network (MANET) is a self-configure LAN that is built spontaneously of devices by using wireless technology. Without using a base station, namely WLAN AP, the communication data flow goes through each node in the network. The individual node receives a data packet, and directly relays it to other nodes. From the viewpoint that a message is delivered across multiple nodes, this ad hoc network is also referred as a multi-hop mesh network. Many researches proposed efficient routing protocols for unstable situation where each mobile node moves freely [85]. Thus, this technology focuses on forming multi-hop network and real-time communication with remotely connected nodes. Furthermore, many data duplication methods for mobile ad hoc network data have been investigated addressing power-aware, partition-aware as well as real-time-aware replications [86]. Our proposal can adopt this temporally formed multi-hop network in the future, but differs from due to the point that the proposal aims to provide temporal storage capability for non-real-time, namely time-offset, communication functionality. The proposal does not require such a route and network construction. However, a new technical issue arises that the duplicative messages are delivered to one terminal many times since message delivery happens at different times and at different places. This paper tackles to reduce network overload brought by this new problem. On the other hand, the opportunistic routing methods that does not need routing are proposed [87]. In this methods, each relay terminal decide whether relay the information to the other relay terminals or not in every situation. The approach is more easier to establish than the predefined routing methods. However, the network load tends to be increased because these network required to stably deliver the target information to the destination terminals.

For time-offset communication, Delay Tolerant Networking (DTN) has been researched [88, 89, 90, 91]. The network has intermittent connections with other networks, and addresses the problem of occasional communications in highly delayed communication situation [92]. DTN is the same as the proposed from the viewpoint that it does not assume a real-time communication as targeted in ad hoc network. Rather, our proposal utilizes DTN technology as a networking technology, and can be categorized into DTN in a sense. The proposed is regard as a special mobile storage platform over DTN technology. For that special purpose, we propose the overall system architecture, and propose a new relay control algorithm to keep store data in a certain storage area while lower overhead and more robust storage capability.

Regarding the researches that store information inside a certain location, [93, 94] proposed a data storing mechanism for a visiting terminal in the future to get information. This method delivers information to connected terminals in an ad hoc matter that are located in a target area. Data distribution is controlled based on the number of hops from the source. They share the similar purpose with our proposal to manage location-related store data. However, since their data distribution and data retrieval are performed on ad hoc network technology mentioned above, the information source terminal distributes the information in a real-time manner. Especially, whereas they perform active

data retrieval, our purpose is that all terminals can passively encounter the store data without requesting or searching store data over network. Namely, every terminal visiting the target storage area can become a relay terminal and a receiver terminal. Then, on receiving relay data, the proposed module inside the terminal can deliver to an appropriate active application the data that it requests. As said before, we aim at simple data sharing by not multi-hop network construction but the direct communication between two terminals. Moreover, we tackle an efficient relay control against new problems brought by thus simple communication style. In addition, since [93, 94], need Global Positioning System (GPS) based location management, it cannot accommodate a wider variety of terminals with no GPS capability. On the other hand, our propose bases on only IEEE802.11 technology to accept various devices like a portable game device or an audio player device.

5.1.3 Difference of the related research

In this section, we describe the related work focusing on the data sharing research for the location related information [94]. This research is mostly similar to our research at the point of our purpose and method. These researches [93, 94] purpose on storing information inside a certain location. This information is delivered only in a target area by terminals in the area using an ad hoc communication. In this research [94], to store the information in the limited area, the terminals relaying information, called "Replica" in the research, are allocated around the target area like surrounding the target area by circle. By these allocation method, a terminal can get the information when it enters the storage area. This consideration is effective to achieve the purpose that all of the terminals passing through the target area obtain the information in the storage area. However, in our research, we do not use any function requiring another calculation cost or process cost like GPS in order to save the power consumption to execute the our algorithm in each terminal. Therefore, it is not appropriate to introduce such allocation technology to our research.

Then, this research also proposes a method to relay the information between the requesting host outside of the target area and the Replica in the target area. In this research, there are the requesting host and it is possible that the requesting host exists outside of the target area because of the possibility that the host want to know the situation of the target area. The approach for effective relay between the requesting host and the target host is skipping the the replica allocation of the location dependent data by using skip parameter. By the control, hosts forwarding the response data store the replica when the hop count meets the determined hop count that represents that the terminal should hold the replication data. This method can decrease the response time and provide high accessibility to location dependent data in the environments where hosts demand the data that is generated at a location far from themselves.

However, the information we consider in our research is located only the target area and this information is useful in the limited area. If other terminal receives the information outside the area, it does not get any advantage to itself. Therefore, no terminal host which requests the information exists outside of the target area in our research. Because of the difference of the handling information, our research require an algorithm which satisfies that all of terminals can receive the information when visits the target area. Moreover, every terminal visiting the target storage area can become a relay terminal in our research, so it is required to reduce the workload for relaying information. Therefore, our purpose is simple data sharing by not multi-hop network construction but the direct communication between two terminals. Then, we tackle an efficient relay control against new problems brought by thus simple communication style.

5.2 Concept of area-based collaborative mobile storage

5.2.1 Proposal basis

We propose a new area-based mobile storage. This storage is formed by distributed multiple terminals, and is maintained by collaboratively sharing and relaying store data among those terminals that pass in/near the local area. Pre-placed storage in the target storage area servers and the Internet infrastructure are not required, and such a storage capability can be provided in anywhere if mobile terminals exit. Even if no terminal stays still in the area, multiple terminals collaboratively store it over time. Additionally, the store data are not propagated over other areas in a borderless way. This enables any applications to leverage this storage platform in anywhere, for example, in people crowded city areas or big exhibition centers.

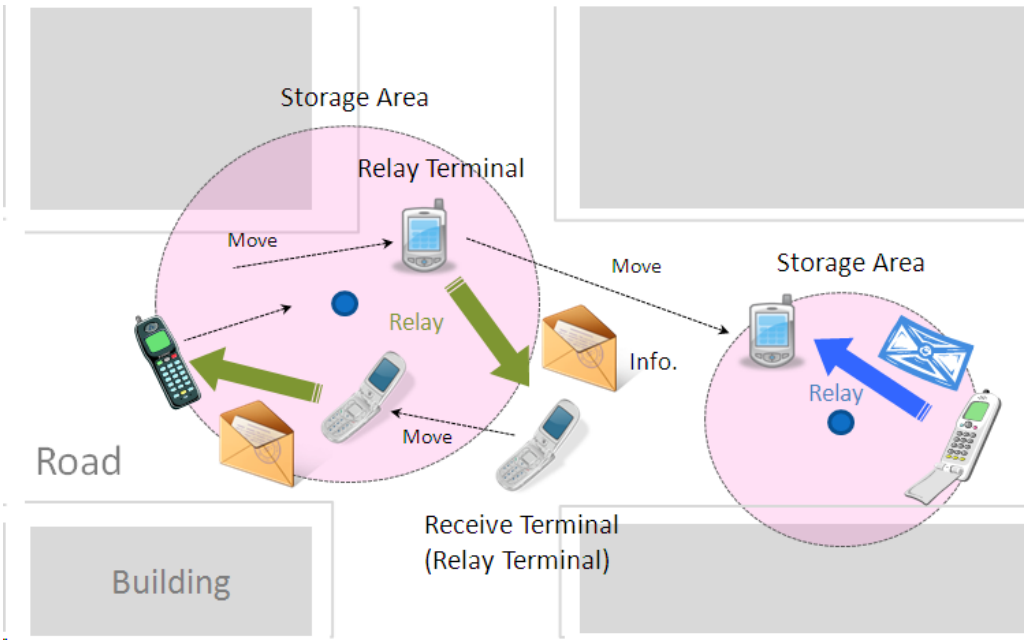


Figure 5.3: Proposal Basis.

Figure 5.3 shows a conceptual behavior in the proposed architecture. Once an end mobile terminal decides to store data in a current area, it stores target store data attached its target storage area information, namely landmark information. Then, it sends the target store data to neighboring terminals. Next, each terminal freely moves inside and outside the storage area, and shares the store data with other terminals. A receiving terminal judges whether the received data should be stored inside the current area or not, and temporally stores and re-shares it with other terminals if needed. Otherwise, the stored data are deleted outside the area. Thus, store data are shared across multiple terminals, and temporally and collaboratively stored inside the storage area, and deleted when outside the area. If a certain visiting terminal needs that store data, it not only relays that data and but also uses it for further services or applications. In addition, an end terminal can be a relay terminal candidate for any store data in any area, and therefore, it might carry multiple store data against multiple storage areas.

5.2.2 WLAN AP-based storage area management

How to recognize location or area in a real physical space is a technically important problem. GPS system is the most powerful tool since it can handle absolute position in any outside location. However, this makes the platform exclude non GPS-equipped, like a handheld portable game, a portable music player, and an ordinal laptop computer. On taking into consideration our original motivation that the proposed platform aims to enhance richer communication services and applications via a short-range wireless communication, we then reached to leverage this communication technology itself to handle the location and area. Namely, detectable WLAN AP can be used to recognize the storage area. It is, however, necessary to emphasize that the WLAN AP information is not utilized for the relay terminals to connect servers or the Internet, but used to detect relative positions against a landmark storage area.

Storage Area Definition

First, the storage area is defined. This paper defines the storage area as a list of the APs that are initially detectable when target data are tried to be stored. Figure 5.4 shows an example of the relationship among store data, storage area, and WLAN APs. In this example, an initial detected three APs when storing data is listed up, and attached to the store data itself. The store data is maintained among terminals while detecting one of three APs.

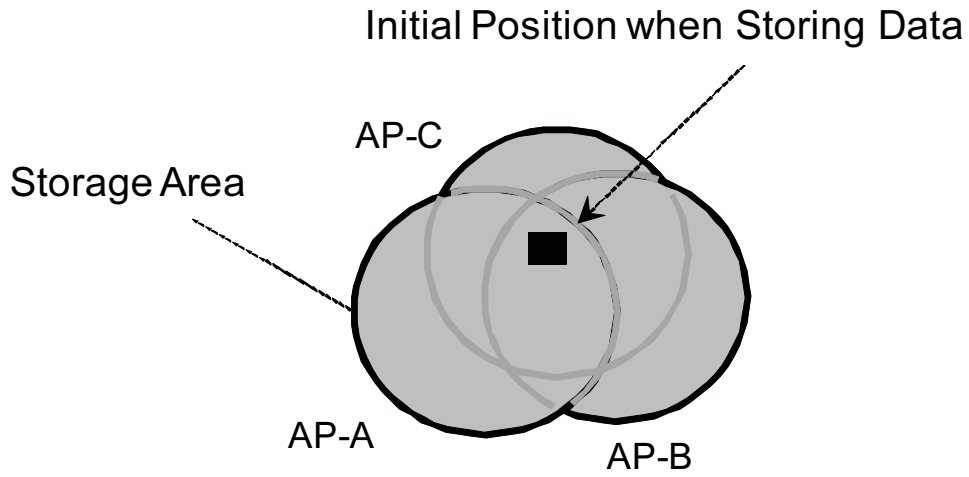


Figure 5.4: WLAN AP-based storage area control.

Area Level and Area Level Table Management

If we could understand positional relation between other AP area and the storage area AP, much finer and more effective relay control would be feasible. This knowledge can be autonomously constructed and updated in each terminal every time a list of detectable terminals changes. In addition, this can be shared among terminals, and collaboratively refined. We use this knowledge in a dynamic relay area control that will be proposed in Section 5.3.3. Here, we introduce a new concept, Area Level, which indicates the relative distance of each AP from the storage area APs. Then, Area Level Table including the level information is created, updated, and shared in and among terminals. More

specifically, we define the distance as the minimal number of AP communication areas to cascade two APs' communication areas. If the two communication areas of two APs overlap, the distance is defined as 0. If they are cascaded by one AP communication area, it is 1. Thus, Level 0 is identical to the base storage area and the neighboring APs of Level 0 AP are categorized into Level 1. In the following, an Area Level Table construction step is explained by using an example depicted as Figure 5.5. Figure 5.5 shows an example of terminal movement through multiple APs from position a (P-a) to position h (P-h). Table 5.1 details the detected APs list at each position, and updating steps of the table that represents the positional relation against AP-A of which is Level 0. A bold italic AP index represents a newly inserted or updated AP.

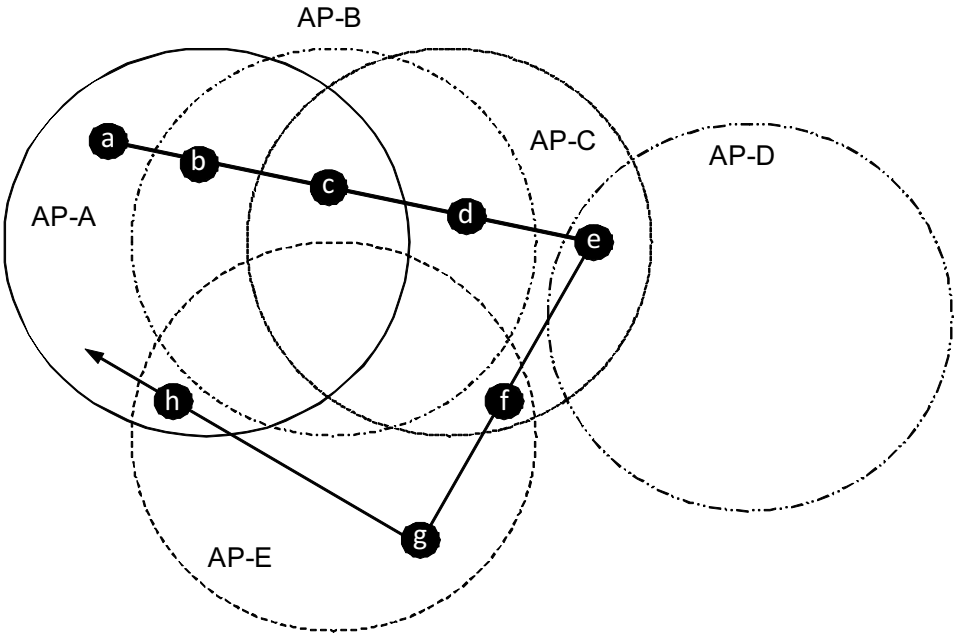


Figure 5.5: WLAN AP-based area level management.

Since AP-A is firstly detected in P-a, AP-A is defined as Level 0. Next, AP-A and AP-B are detected in P-b, and AP-B is inserted into the table as Level 1 because it is detectable together with AP-A at the same time. This means that each communication areas of AP-A and AP-B overlap. Similarly, AP-C is added into Level 1. AP-D detected with AP-C in P-e is inserted into Level 2 since AP-C is listed in Level 1. In P-f, AP-E is similarly inserted into Level 2. The point is that AP-E is modified into Level 1 in P-h because AP-A and AP-D are detectable here and AP-A is, of course, listed as Level 0. Although this example describes a creation and update of the table inside one terminal, an integration of two separate tables can be achieved. Level inconsistency against one AP can be resolved by selecting a lower level.

5.2.3 Platform architecture overview

This section describes an overview of our proposed platform. The collaborative mobile storage module is located inside each terminal. It can be implemented in a middleware separated from 3-rd party applications, or might be combined to each application itself. The platform architecture is depicted in Figure 5.6.

Table 5.1: AREA LEVEL TABLE CONSTRUCTION STEP EXAMPLE.

	a	b	c	d	e	f	g	h
Detect AP	A	A,B	A,B,C	B,C	C,D	C,E	E	A,E
Level 0	A	A	A	A	A	A	A	A
Level 1		B	B,C	B,C	B,C	B,C	B,C	B,C,E
Level 2					D	D,E	D,E	D

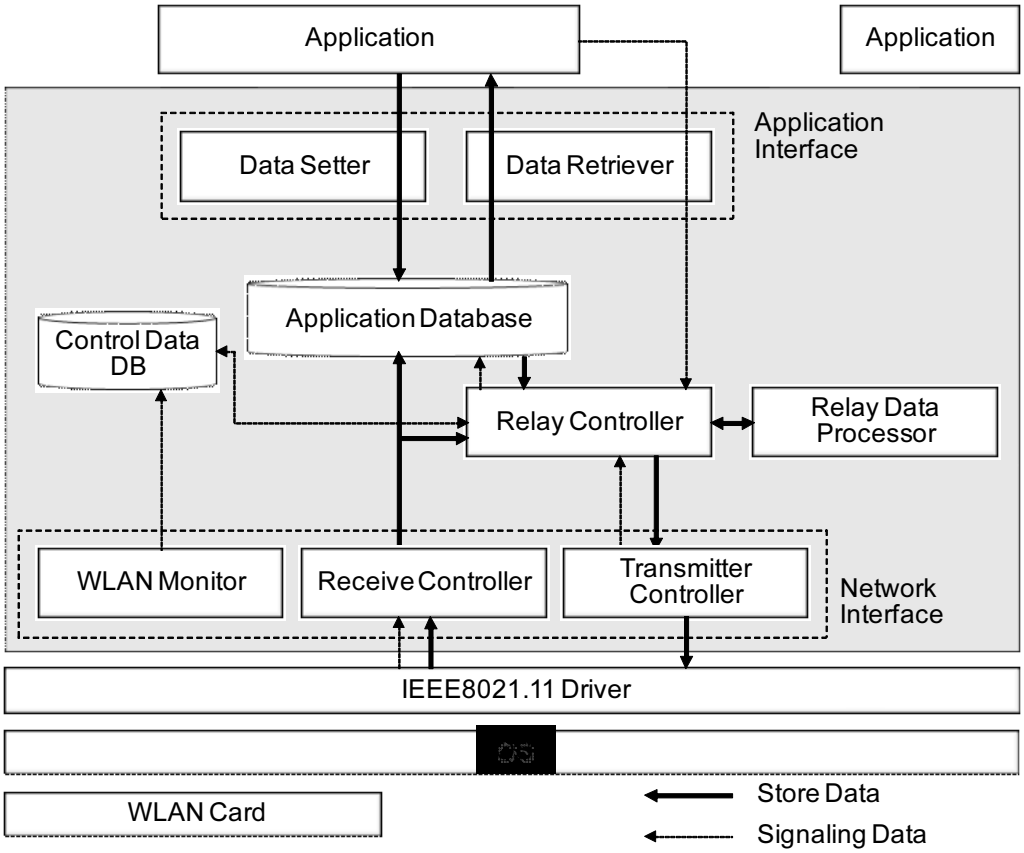


Figure 5.6: Platform architecture.

The central function is composed of the Relay Controller, Application Database, Relay Data Processor, and Control Data Database. The Relay Controller is the core engine to determine when and to which to send store data. It also controls the size of the storage area. These algorithms are our main contribution, and are described in Section 5.3. The Application Database temporally accumulate received multiple store data with meta data for the relay control. The meta data includes the storage area information mentioned in the previous subsection, terminal id of which has the corresponding store data, received timestamp, target application id, data type like text or binary data, and so on. The Relay Data Processor provides a multiplex function of multiple different store data to be sent to other terminals. It also de-multiplexes and parses received data to store to the Application Database. The Control Data Database manages various information accessing WLAN driver modules. It includes currently detectable WLAN AP information, i.e. SSID, and neighboring terminal information, MAC address, inside an ad hoc communication range.

In addition, the Application Interface and Network Interface are equipped. The Application Interface provides a proxy between the core functions mentioned above and applications. The applications read and write store data via the Application Data Retriever/Setter interface from/to the Application Database. When writing data to the database to be stored and relayed, the Data Setter retrieves currently detectable AP list, and attaches it with the store data payload. For reading, stored data are delivered to corresponding applications based on its application id. Meanwhile, regarding the Network Interface, the WLAN monitor retrieves detectable WLAN AP Extended Service Set Identifiers (ESSIDs) in an infrastructure mode and end terminal Media Access Control (MAC) addresses in an ad hoc communication mode, and inserts them into the Control Data database. The Receive Controller and Transmitter Controller provide proxy interfaces against the core functions for the store data to be relayed.

5.2.4 System behavior

Let us start from when one application tries to start store data to this platform. The data are firstly stored in the Application Database via the Data Setter. Meta data are attached with the store data, which is its application id, its data type, the current timestamp, and the list of the currently detected APs as target storage area information. The list of APs is already handled in Control Data Database. On the other hand, the Relay Controller periodically checks whether store data to be relayed exist or not, and calculates a relay timing and terminal to share against each store data. The relay timing and terminal are completely different in multiple store data because they have their individual pass route, received timing, storage area, and density of terminals that already have. If more than one store data should be relayed, one multiplexed data created in the Relay Data Processor is relayed to another terminal via the Transmitter Controller.

Once a terminal receives relayed store data to be relayed via the Receiver Controller, the Relay Controller judges whether each component relayed data should be stored into the Application Database or not by parsing the de-multiplexed relay data from the Relay Data Processor. Each component relay data are stored as independent different store data, and informed into corresponding application via the Data Retriever.

Separately, the Relay Controller determines whether each store data should be kept or not every time detected APs change. Once the terminal leaves a storage area of a certain store data, the data are deleted from the Application Database, and not relayed any more from that terminal.

5.3 Details of relay control algorithm

5.3.1 Problems and requirements for relaying store data

A length of time period while two moving terminals can communicate in an ad hoc communication mode is quite short. Therefore, as the same as existing practical applications, this paper assumes a simple communication protocol, where relay terminal unidirectionally sends store data to other terminals without querying in advance whether its corresponding terminal has the store data to be relayed or not.

We have two problems in our mind to investigate an efficient mobile storage. The first is that a flooding-based information sharing causes the load of communication overhead between terminals, and duplicative delivery of the same store data to the same terminal. Therefore, it is needed to reduce such a network overhead while keeping storage time as longer as possible. We propose in Section 5.3.2 a new relay timing and terminal determination algorithm based on an estimated data-holding ratio inside reachable terminals.

The other issue is that casual sparse density of the terminals in a target storage area causes lower storage capability. Even if a sufficient number of candidate terminals exist close to the storage area, the sparse terminal density inside the storage area cannot steadily keep store data. Therefore, it is necessary to reduce disadvantageous influences brought by nonuniformly distributed terminals, and give a more steady and stronger storage capability. A new relay area control based on a terminal density in the storage area to tackle this problem is proposed in the following Section 5.3.3.

5.3.2 Relay timing and terminal determination algorithm

Before the detailed explanation, we define two states of a terminal against store data, that is data-holding state and unknown state. The former indicates that the terminal has already the target store data, and the latter means that it is unknown whether the terminal has the data or not.

This algorithm aims for each terminal to determine when and to which terminal to relay store data. The terminal performs the following calculation every pre-defined time, and sends store data to other terminals if needed. The basic idea is that each terminal autonomously understands which terminal has no a target store data to be relayed, and estimates the self-responsible number of the destination terminals to be relayed. Figure 5.7 illustrates an example of positional relation and data-holing state of multiple terminals. A target terminal, T-X, calculates when and to which terminal to send the received store data that were relayed from T-a. This calculation is resolved based on a list of data-holding terminals and shared statistical knowledge. The former provides the number of detectable data-holding terminals, nhold. This can be created by a list of relaying terminals (T-a, T-f) that in past sent the target data to the terminal, and a list of data-holding terminal ids (T-b, T-h, T-g) that was received accompanied with the store data payload from the relaying terminal. These give the calculating terminal sure knowledge which detectable terminals have already the target data. The latter is used to estimate the number of data-holding terminals, e_{hold}^{unk}, out of all unknown-state terminals (T-c, T-d, T-e), n_{all}^{unk}, except the data-holding terminals mentioned above. This is the previously used statistic data that was relayed accompanied by the target store data. The statistic data includes the combination of the number of all detected terminals, mall, and data-holding terminals, mhold, which the previous relaying terminal calculated. According to this calculated number, we can relay the store data to the estimated non-data-holding terminals while keeping duplicative relays as less as possible. Table 5.2 summarizes all of notation used in the proposed algorithm. Note that they are for the terminals inside an ad hoc communication range.

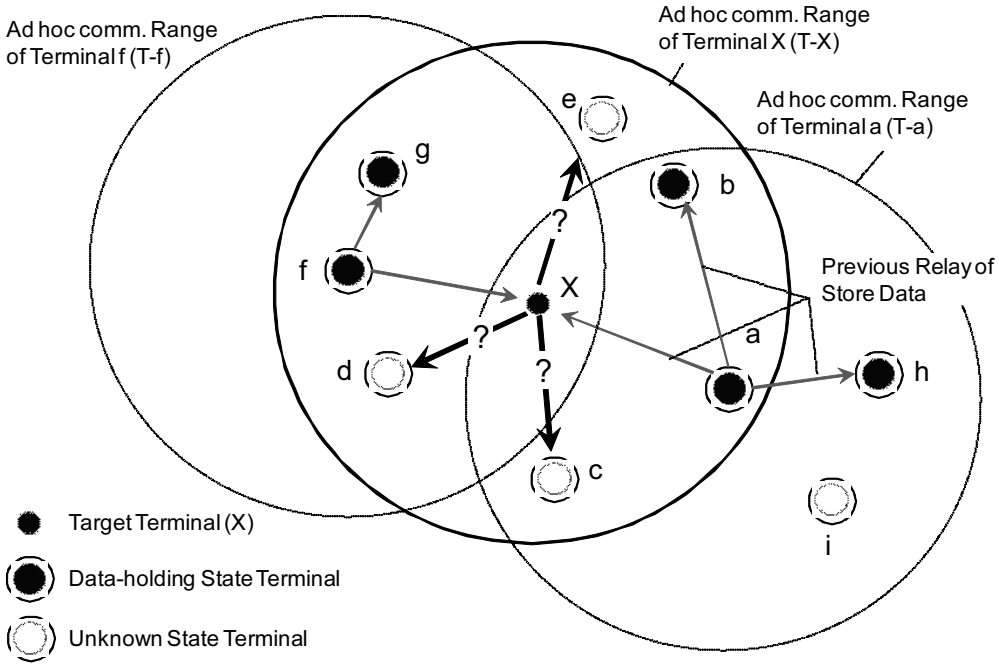


Figure 5.7: Relay timing and terminal determination algorithm.

First, the estimated number of data-holding terminals in the unknown state terminals, e_{hold}^{unk}, is calculated in (5.1).

$$e_{hold}^{unk} = n_{hold}^{unk} \times \frac{m_{hold}}{m_{all}} \tag{5.1}$$

where, n_{all}^{unk} is the number of the unknown-state terminals inside the area that is detectable by an ad hoc communication mode. mall and mhold are the previous total number of all terminals, and the previous data-holding terminals, respectively. Then, the estimated number of all data-holding terminals, e_{hold}, is calculated by adding e_{hold}^{unk} into the number of data-holding terminals, n_{all}^{unk}, as shown in (5.2).

$$e_{hold} = n_{hold} + e_{hold}^{unk} \tag{5.2}$$

The number of terminals to be relayed, l_{relay}, is calculated by dividing the estimated number of non-data-holding by all of data-holding terminals including this terminal. This is represented in (5.3).

$$l_{relay} = fracn_{all} - e_{hold}n_{hold} + 1 \tag{5.3}$$

Finally, this terminal selects the closest l_{relay} terminals, and relays the store data to those terminals. Note that we assume that the ratio of inaccessible terminals from the data-holding terminals inside are approximately equal to that of accessible from non-detectable other terminals outside area to the estimated data-non-holding terminals.

Table 5.2: NOTATIONS AND DESCRIPTION USED IN PROPOSED ALGORITHM.

Notation	Description
n_{all}	Number of detectable all terminals.
n_{hold}	Number of detectable data-holding terminals.
n_{all}^{unk}	Number of detectable unknown-state terminals.
e_{hold}^{unk}	Estimated number of data-holding terminals in unknown-state terminals.
e_{hold}	Estimated number of all data-holding terminals.
m_{all}	Previous number of detectable all terminals.
m_{hold}	Previous number of detectable data-holding terminals.
l_{relay}	Number of terminals to be relayed

5.3.3 Relay area control algorithm

This algorithm controls the size of the storage area in order to reduce disadvantageous influences brought by nonuniformly distributed terminals, and to give more steady and stronger storage capability. Since fewer terminals bring lower storage capability, this platform extends the storage area into its surrounding areas to increase the number of relay terminals. Here, we redefine the initial storage area as base storage area, and the extended area is newly defined as expansion area. The storage area finally is composed these base and extension areas. The concept of our relay area expansion is depicted in Figure 5.8.

A basic and simple algorithm proposed in this paper is as follows. Since the storage capability depends on the density of terminals in the area, a judgment of area expansion is achieved by comparing an estimated terminal density with a pre-defined threshold. If the density is lower than the threshold, the area is expanded since the storage capability goes down. This expansion is autonomously controlled by each terminal. As for the terminal density, we can estimate it from history data of the measured detectable terminals during a certain past period. In addition, since such monitoring data are restricted to the terminal's small knowledge, we can utilize the previous number of detectable all terminals, mall, that is received from other terminals. An optimum calculation of the threshold is a future work. Note that this judgment is performed every time detectable APs list changes.

Once an area expansion is determined, its expansion area information, i.e. AP information, and its new expansion level information are newly updated to the storage area information into the Application Database inside the terminal, and relayed to others accompanied with store data. These follow the Level representation and the Area Level Table described in Section 5.2.2.

Shrinking an expansion area is slightly difficult. The current expansion level and table give a level of a newly detected AP. When the current area goes to a lower level area, the current expansion level is simply returned to the lower level, and the same judgment mentioned above is performed subsequently.

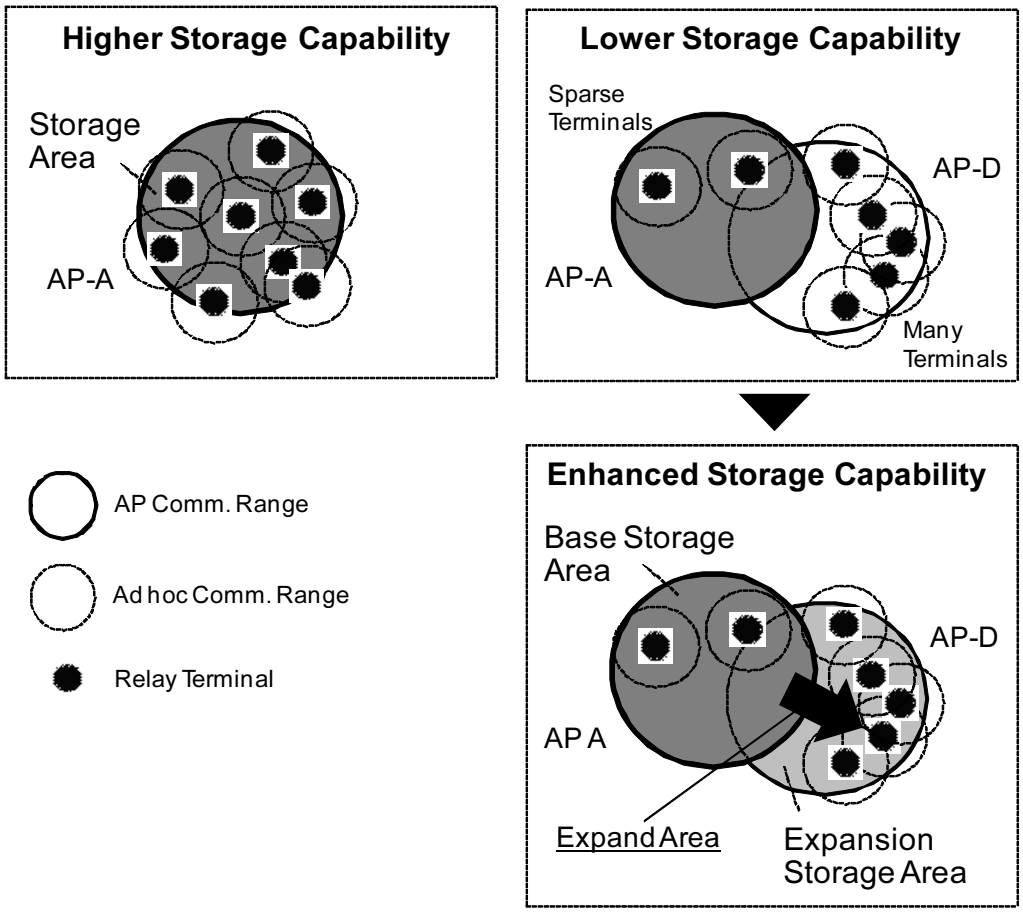


Figure 5.8: Concept of Relay Control Area Expansion.

5.4 Simulation evaluation

5.4.1 Simulation method

We developed a simulator that can simulate a terminal mobility model and implement the proposed relay control algorithm in the simulator. Mobile terminals move based on a mobility model. Figure 5.9 shows the screenshot of the simulator. It is implemented in Java and and it can visualize the state of each terminals during the simulation. Here, the details of simulator and the settings are described as follows.

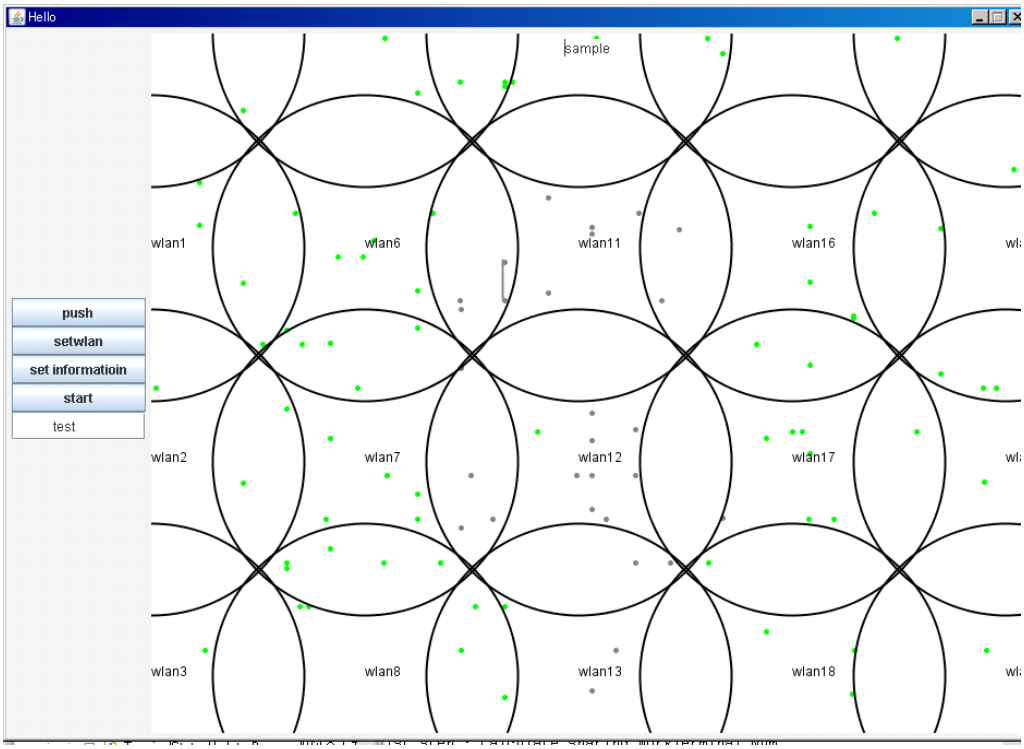


Figure 5.9: Simulator screenshot.

As for an AP setting, multiple APs along the lines can be also set on the space. Based on the configured size of AP communication range, multiple APs are placed closely together at a preconfigured regular interval. With respect to the terminal generation model, according to the configurable number of terminals inside the simulation area, terminals appear from grid point based on two occurrence models; the random model and the Poisson distribution model. This generation is kept to make the number of active terminals on the space constant. If one terminal disappears from this space, another new terminal is generated from randomly selected cross point. Next, regarding the terminal mobility model, a mobile terminal changes its moving direction on each cross point of lines as described in Figure 5.10. The mobile terminal selects randomly one direction among three directions excluding its current reverse direction. In addition, it randomly pauses on the way. On each running step, each terminal selects its moving behavior. Table 5.3 shows the simulation parameter used in this simulation.

Finally, as a graphical interface, on the simulation virtual 2D space, the lines where terminal moves on follow a grid pattern as shown in Figure 5.10. The graphical animator draws the WLAN APs communication range as a static circle, and shows the behavior of mobile terminals visually,

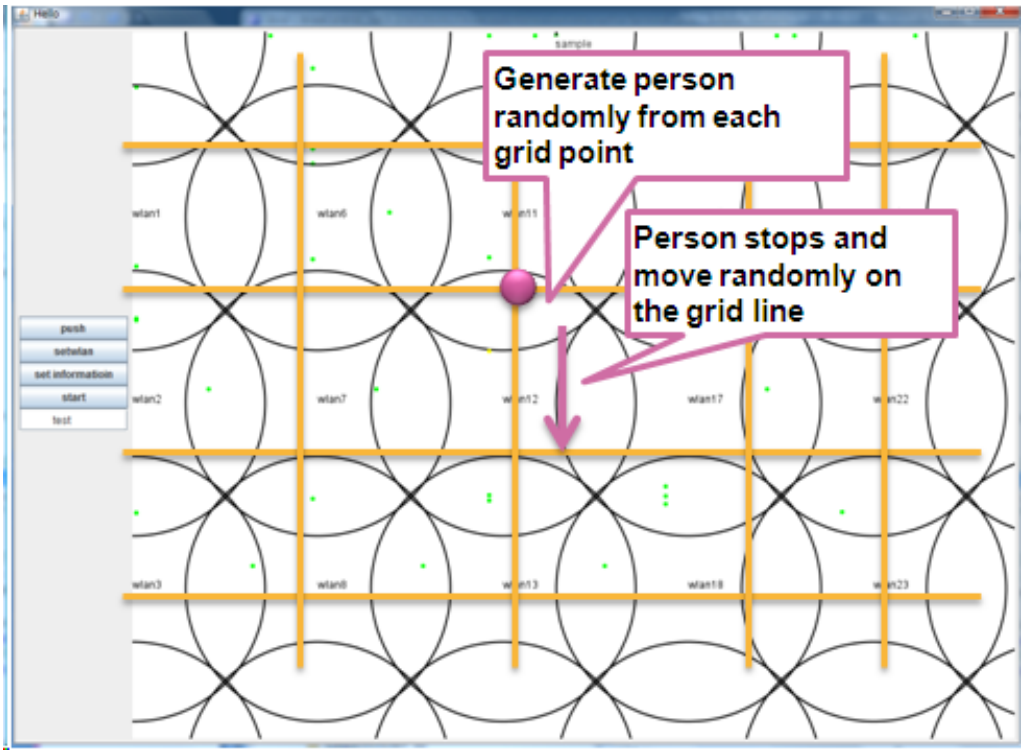


Figure 5.10: Simulator screenshot and model concept.

indicated by small moving points, based on simulation results. A data relay operation is also drawn in line that connects two connecting terminals.

5.4.2 Experiment results

Performance Evaluation for Timing and Terminal Determination Algorithm

This section preliminarily evaluates a performance of the proposed relay timing and terminal determination algorithm in Section 5.3.1 . A comparison method is a flooding-based method, where a terminal sends store data to any terminals within an ad hoc communication range. To set comparative two methods' relay capability outside the storage area, the relay operation in the flooding-based method is not performed outside the storage as the same as the proposed method. The purpose of

Table 5.3: CONFIGURATION PARAMETERS IN SIMULATION.

Parameter	Value
WLAN AP Communication Range	70 (m)
Ad hoc communication Range	20 (m)
Human Waling Speed	1.0 (m/s)
Simulation Space	400 (m) x 320 (m)
AP interval distance	93 (m) (=70*2*2/3)
Number of total WLAN APs	20
Simulation time duration	60 (minutes)

Table 5.4: NOTATIONS AND DESCRIPTION USED IN EVALUATION.

Notation	Description
N_{all}	Number of passing terminals in storage area.
N_{hold}	Number of data-holding terminals in storage area.
R_{hold}	Ratio of data-holding terminals against out of all.
N_{rece}	Number of data-received terminals when leaving storage area.
R_{rece}	Ratio of data-received terminals when leaving storage area.
N_{relay}	Number of relay operations.
N_{dup}	Number of duplicative relays.
T_{store}	Time length of store time.

this evaluation is that the proposed method can reduce the number of duplicative relays, N_{dup}, while preserving data receiving ratio higher. This higher ratio, R_{rece}, represents more passing terminals can receive the store data when passing through the storage area. The number of active terminals, N_{active}, is set as 70 in this experiment. Table 5.4 shows the metrics to be evaluated in this simulation.

Figure 5.11 shows the number of all terminals, N_{all}, and data-holding terminals of the conventional and proposed method, N_{hold}. Figure 5.12 also shows the ratio of data-holding terminals out of all terminals, R_{hold}. Figure 5.13 shows the accumulated number of data-received terminals out of all, N_{rece}. From these, we can conclude that two methods can give almost the same performance although the conventional method is sometimes slightly better.

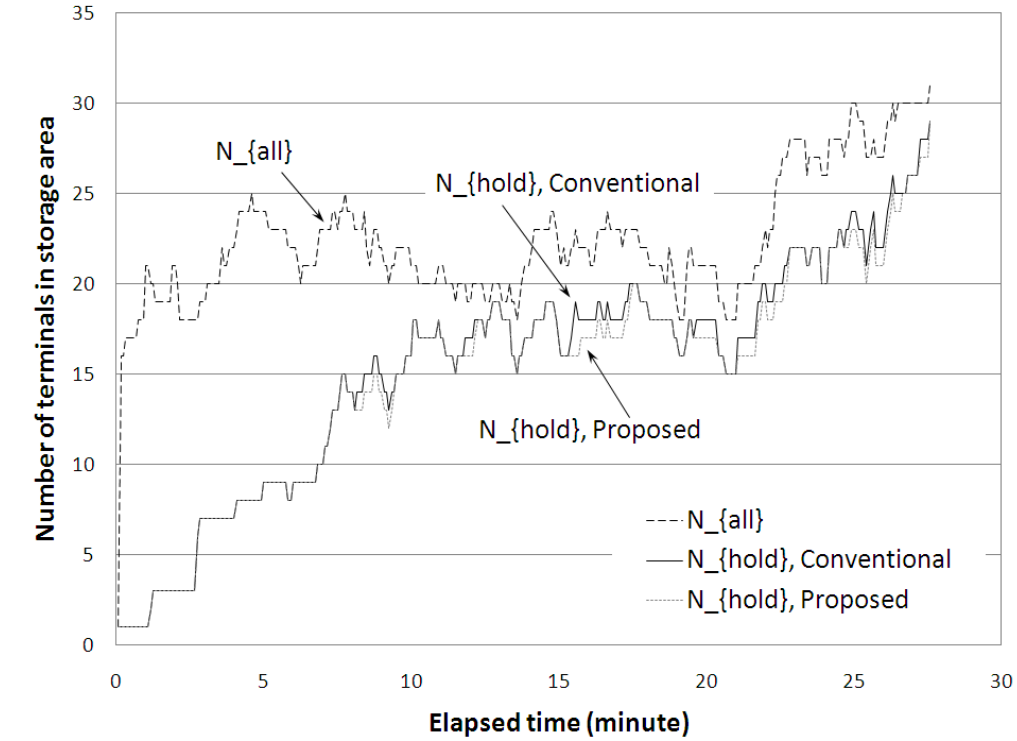


Figure 5.11: Number of terminals (N_{all}) and data-holding terminals (N_{hold}).

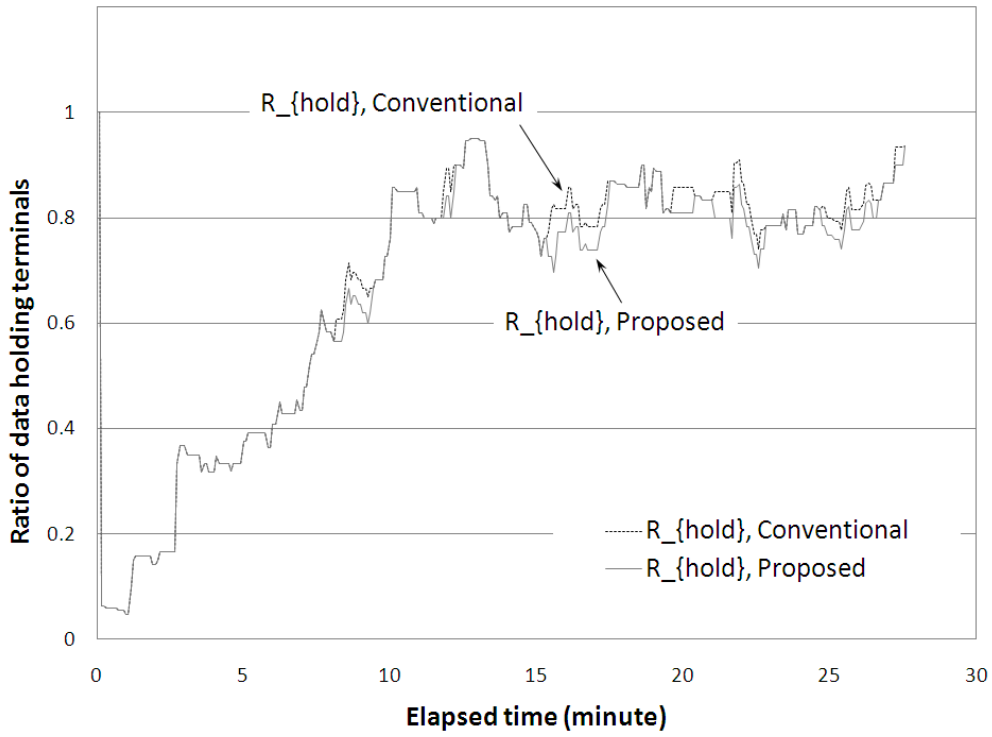


Figure 5.12: Ratio of data-holding terminals out of all (R_{hold}).

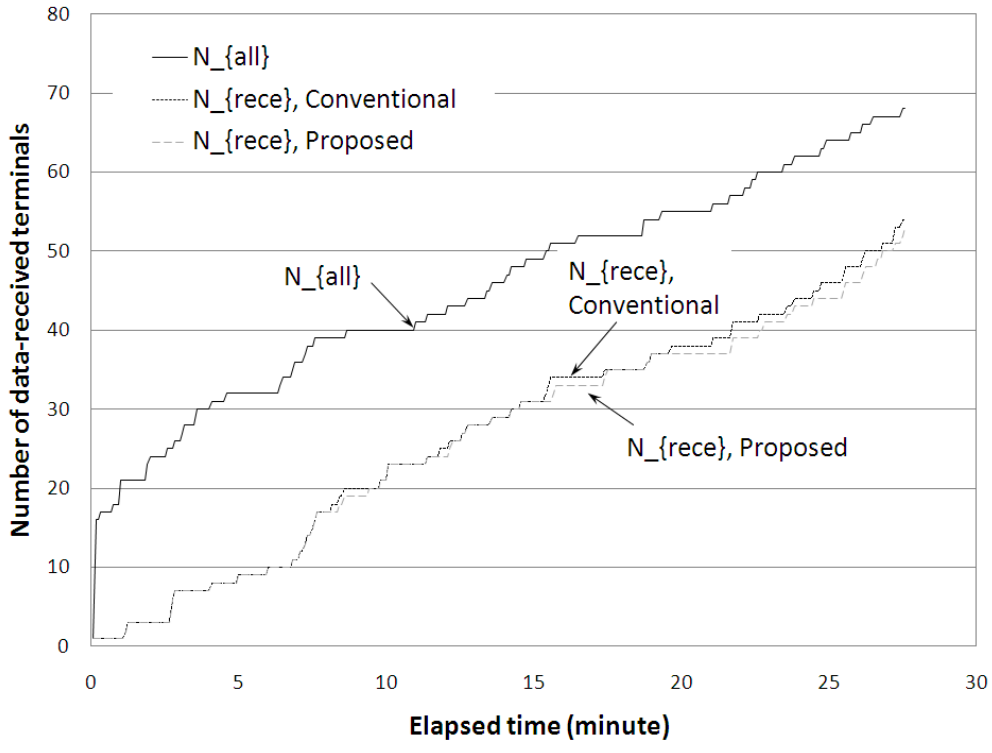


Figure 5.13: Accumulated number of data-received terminals out of all (N_{rece}).

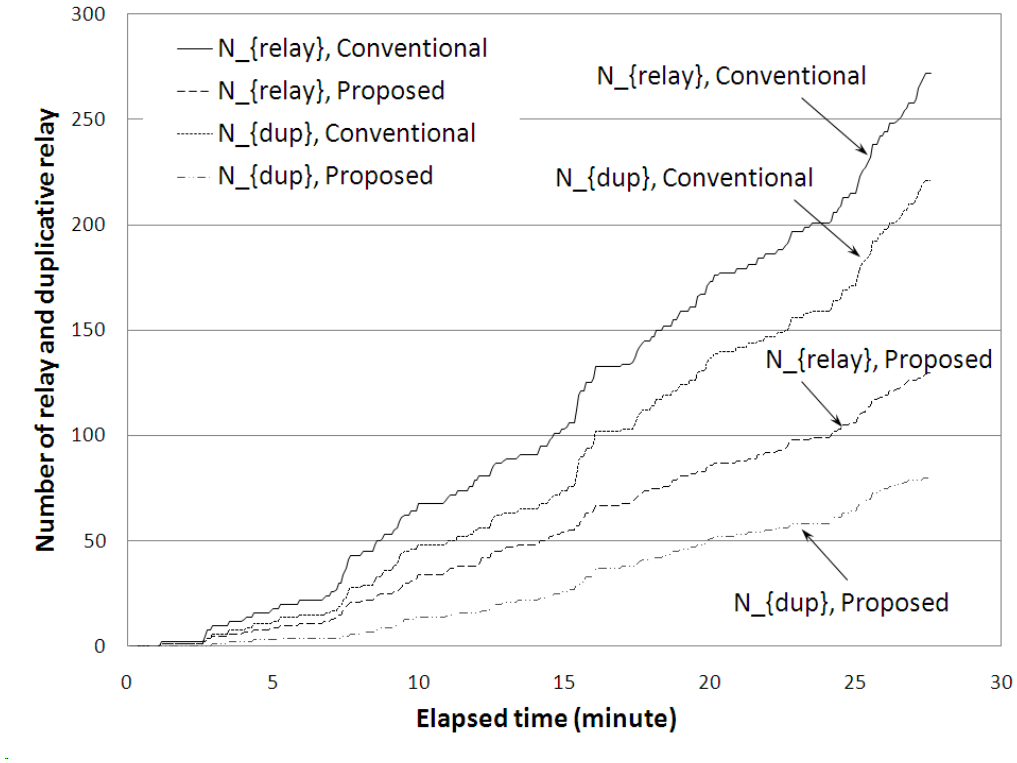


Figure 5.14: Accumulated number of relay (N_{relay}) and duplicative relay (N_{dup}).

Next, Figure 5.14 shows the accumulated number of relays, N_{relay}, and duplicative relays, N_{dup}. The proposed method can drastically reduce the number of relay operations and duplicative relays compared with the conventional method. The more dense terminals is supposed to lead to more overloaded network. As a result, the proposed method can provide the same storage capability while achieving significant network overhead reduction compared with the flooding-based method.

Figure 5.15 shows the number of all terminals, N_{all}, and data-holding terminals of the conventional and proposed method, N_{hold}. Figure 5.16 also shows the ratio of data-holding terminals out of all terminals, R_{hold} when N_{active} is set 50. Figure 5.17 shows the accumulated number of data-received terminals out of all, N_{rece}. From these, we can conclude that two methods can give almost the same performance although the conventional method is sometimes slightly better.

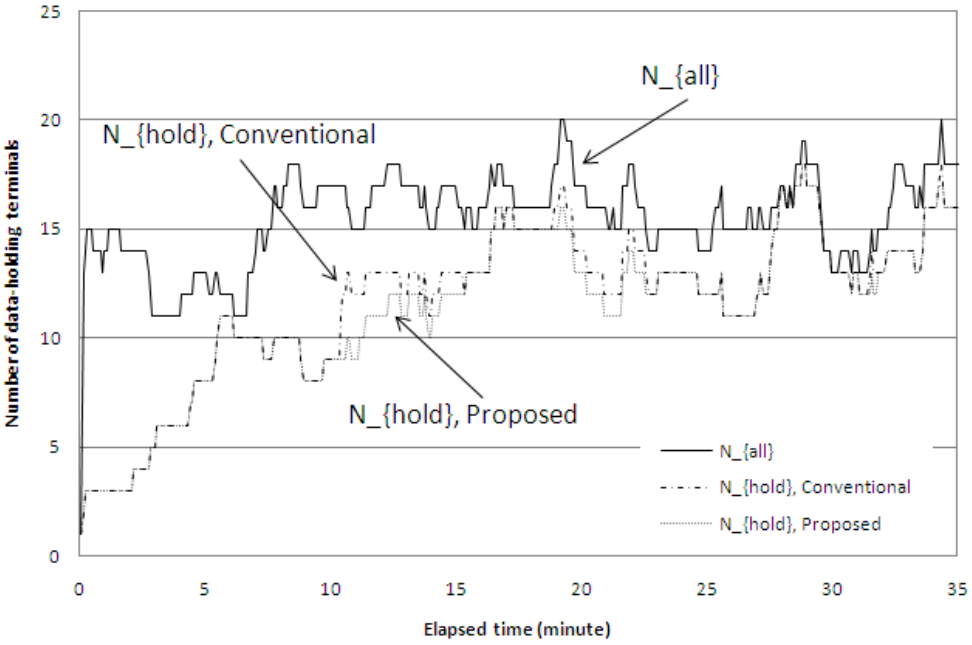


Figure 5.15: Number of terminals (N_{all}) and data-holding terminals (N_{hold}) (N_{active}=50).

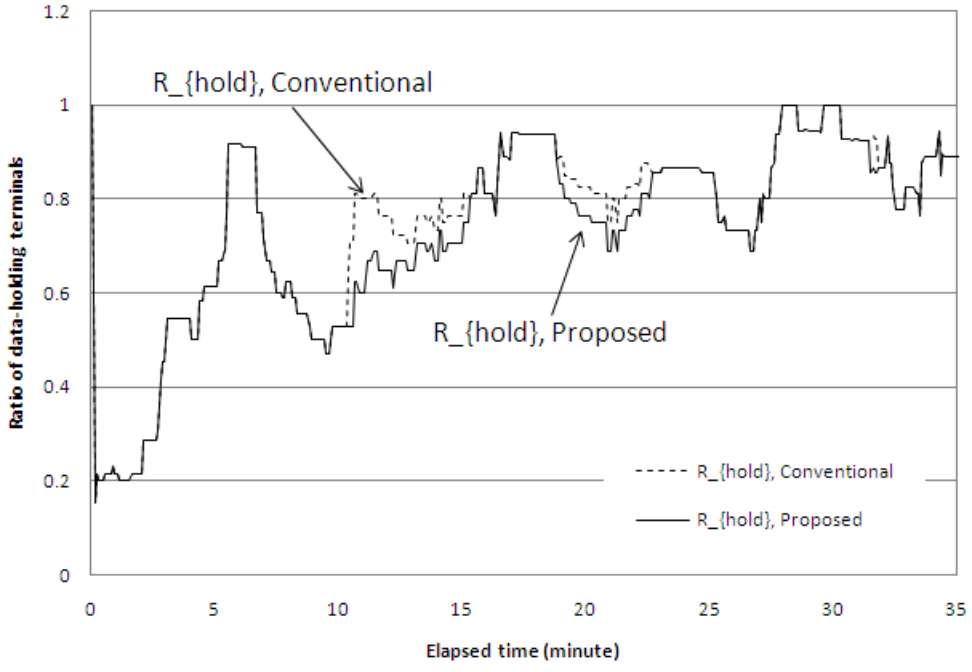


Figure 5.16: Ratio of data-holding terminals out of all (R_{hold}) (N_{active}=50).

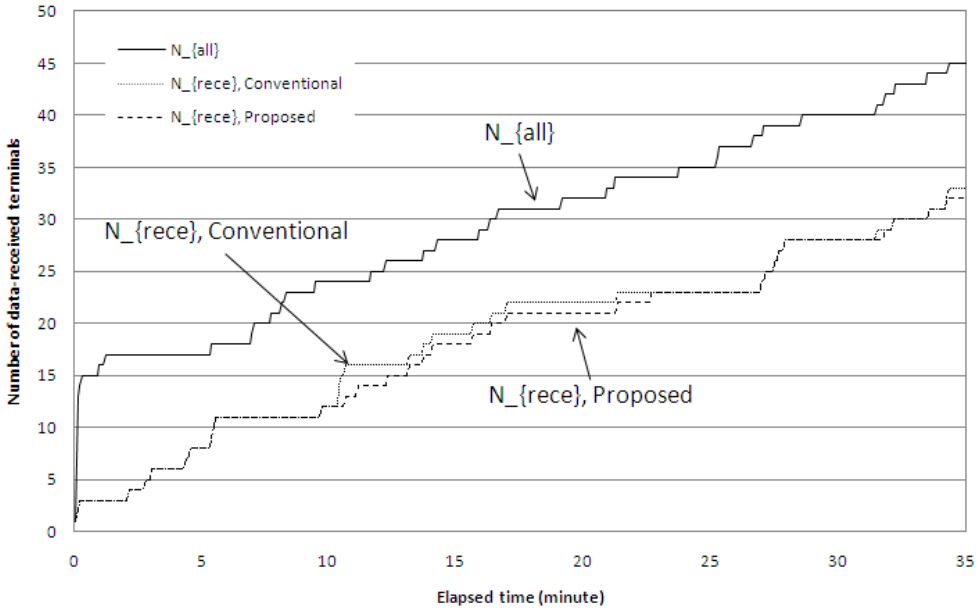


Figure 5.17: Accumulated number of data-received terminals out of all (N_{rece}) (N_{active}=50).

Next, Figure 5.18 shows the accumulated number of relays, N_{relay}, and duplicative relays, N_{dup}. The proposed method can drastically reduce the number of relay operations and duplicative relays compared with the conventional method. The more dense terminals is supposed to lead to more overloaded network. As a result, the proposed method can provide the same storage capability while achieving significant network overhead reduction compared with the flooding-based method.

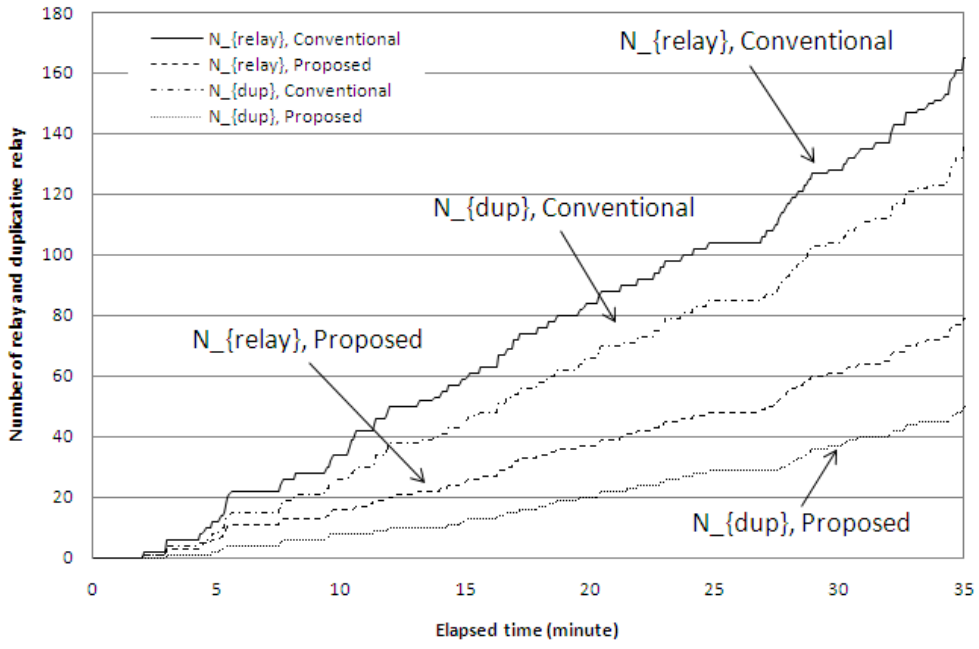


Figure 5.18: Accumulated number of relay (N_{relay}) and duplicative relay (N_{dup}) (N_{active}=50).

Figure 5.19 to 5.22 show the results when N_{active} is set 40.

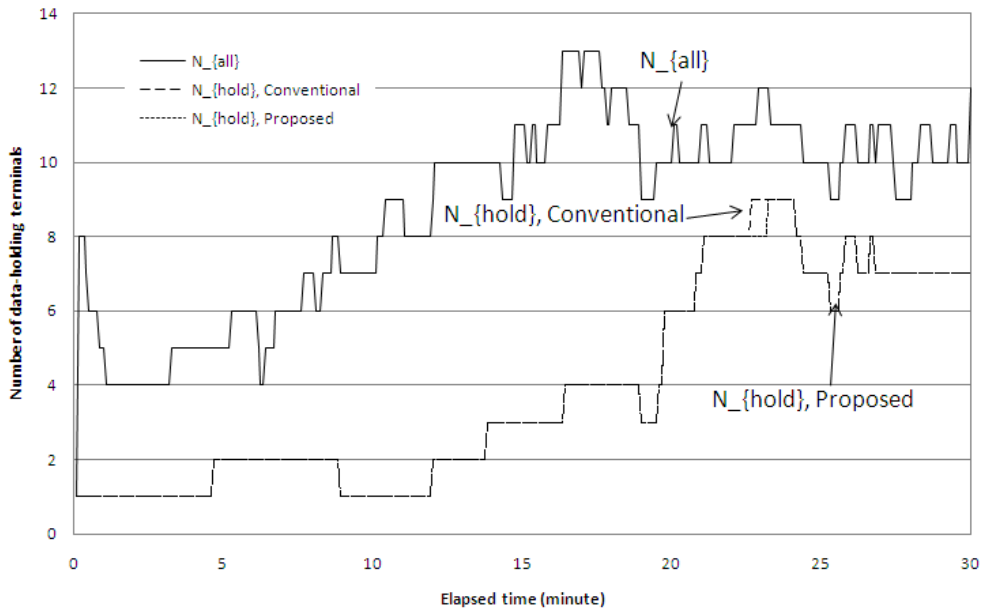


Figure 5.19: Number of terminals (N_{all}) and data-holding terminals (N_{hold}) (N_{active}=40).

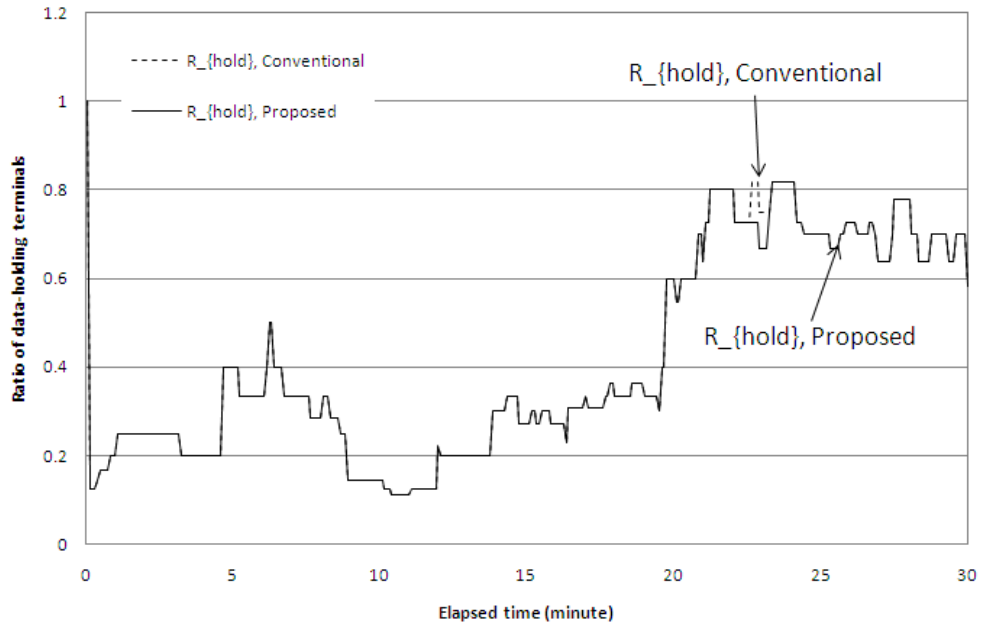


Figure 5.20: Ratio of data-holding terminals out of all (R_{hold}) (N_{active}=40).

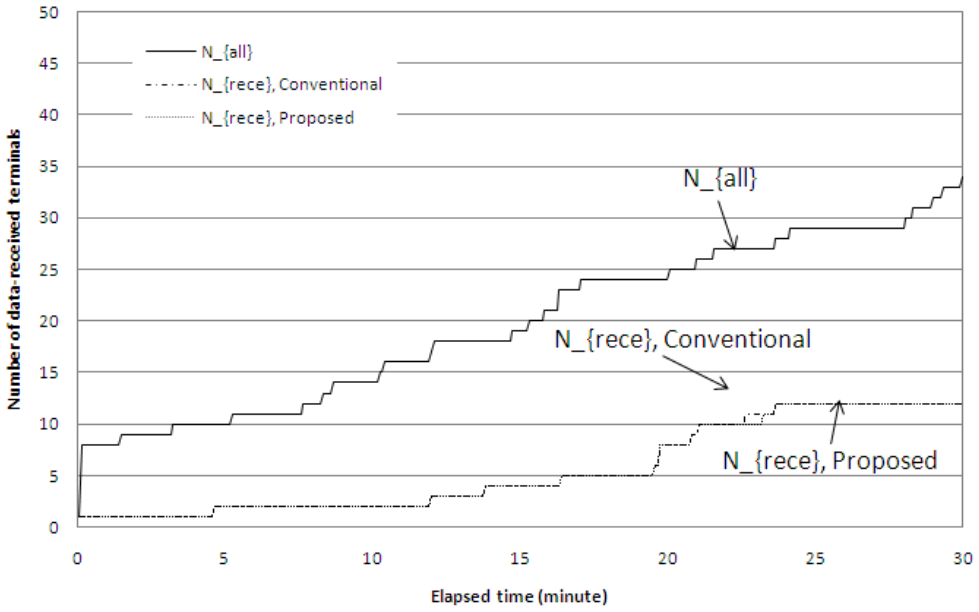


Figure 5.21: Accumulated number of data-received terminals out of all (N_{rece}) (N_{active}=40).

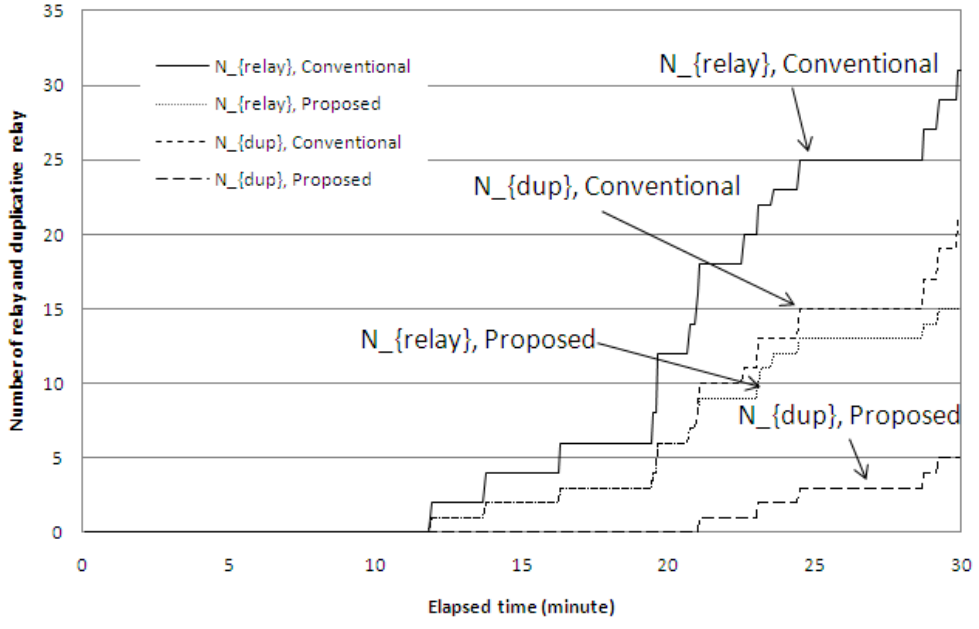


Figure 5.22: Accumulated number of relay (N_{relay}) and duplicative relay (N_{dup}) (N_{active}=40).

Figure 5.23 to 5.26 show the results when N_{active} is set 30.

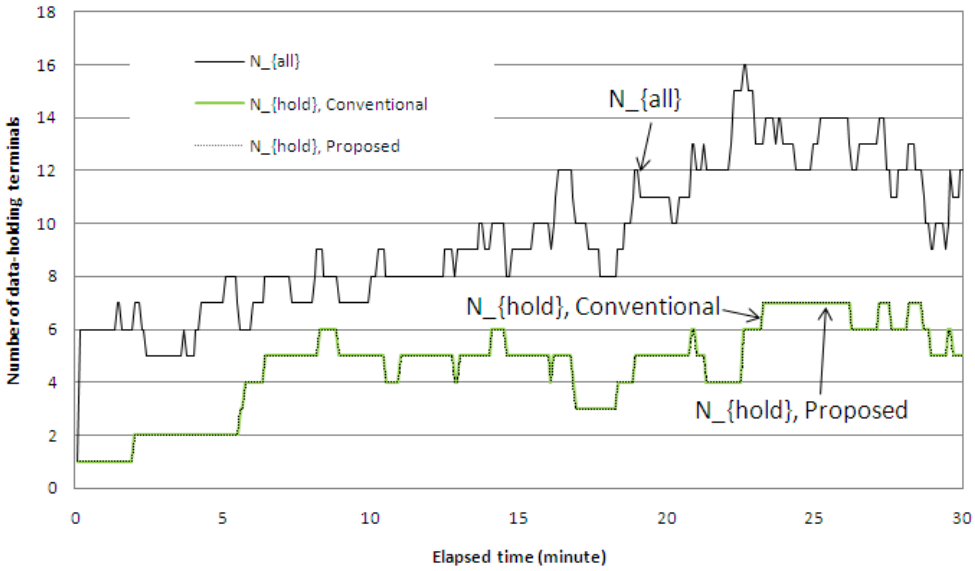


Figure 5.23: Number of terminals (N_{all}) and data-holding terminals (N_{hold}) (N_{active}=30).

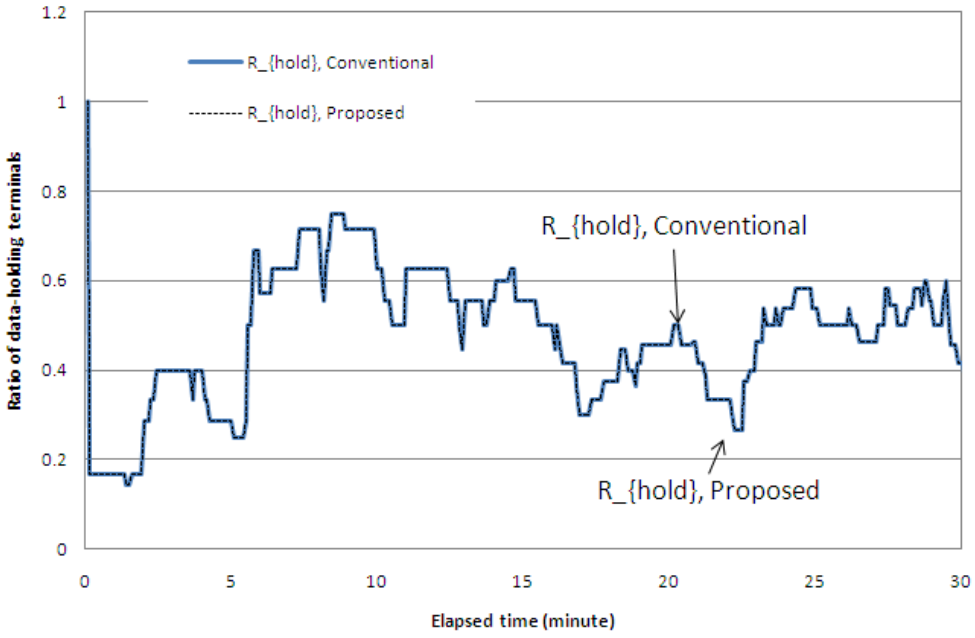


Figure 5.24: Ratio of data-holding terminals out of all (R_{hold}) (N_{active}=30).

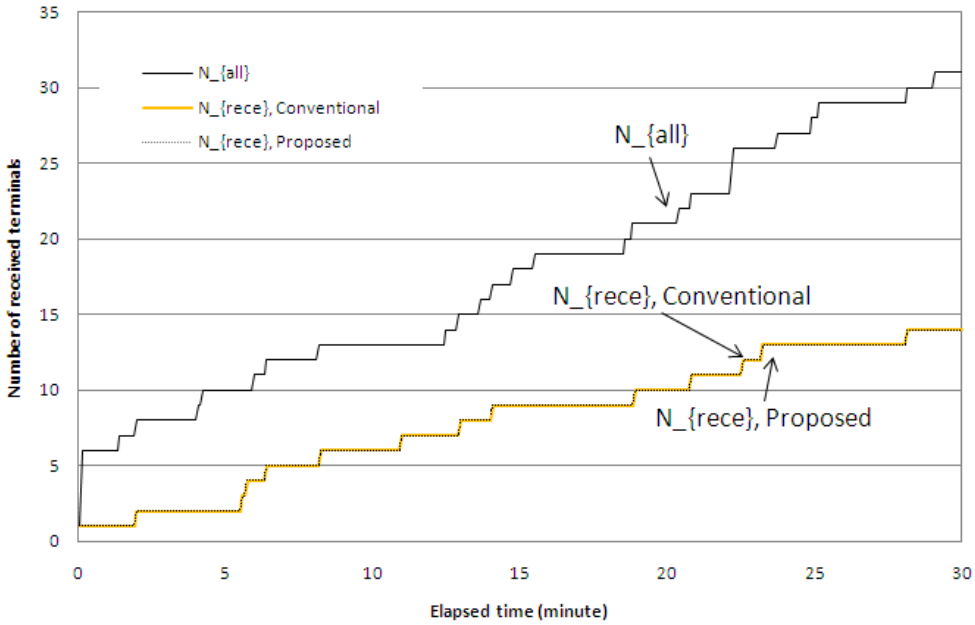


Figure 5.25: Accumulated number of data-received terminals out of all (N_{rece}) (N_{active}=30).

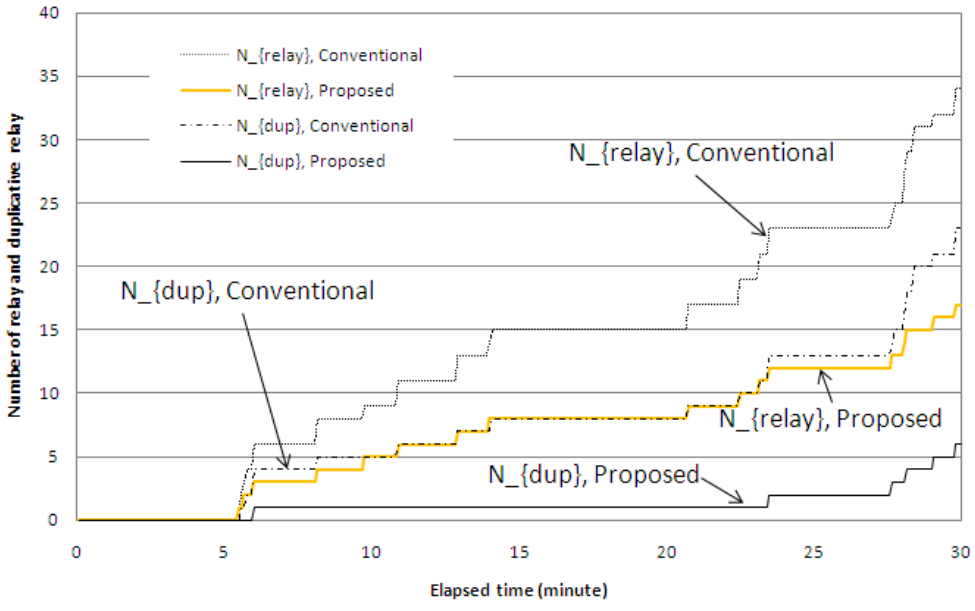


Figure 5.26: Accumulated number of relay (N_{relay}) and duplicative relay (N_{dup}) (N_{active}=30).

Considering these results when N_{active}=30, the lower than N_{active} is our proposed algorithm decide the relay timing whenever terminal finds another person because there are few people to store the information in the storage area.

So, to store the information for a long time as far as possible, our proposed algorithm has the relay area control. In the next session, we evaluate the area control algorithm in detail.

Performance evaluation for relay area control

The evaluation experiments for the proposed relay area control described in Section 5.3.3 are conducted. We compare two methods; Proposed A and Proposed B. The former is identical to the method with the timing and terminal determination algorithm. The latter is newly added the proposed area control to Proposed A. In this simulation, the length of storage time, T_{store}, is measured as the metric representing the storage capability. We show that the proposed area control can extend T_{store} even if the number of relay terminals in the storage area decreases. In this simulation, we select two overlapped APs as the base storage area. This means that this has possible 10 APs as the expansion storage area. Hereafter, to distinguish which area each measured number indicate, namely base area or expansion area, superscript notations are applied like N_b and N_e for the number of base area and expansion area, respectively.

Figure 5.27 shows the relation between T_{store} and the number of active terminals, N_{active}. N_{active} is 5, 10, 15, and 20. Note that Proposed B did not reach an end when N_{active} is more than 20. This result is an average value in 10 times experiments per each case. As expected, T_{store} in both methods decreases as N_{active} decreases. However, Proposed B can maintain overwhelming higher storage capability over Proposed B thanks to support from the neighboring terminals.

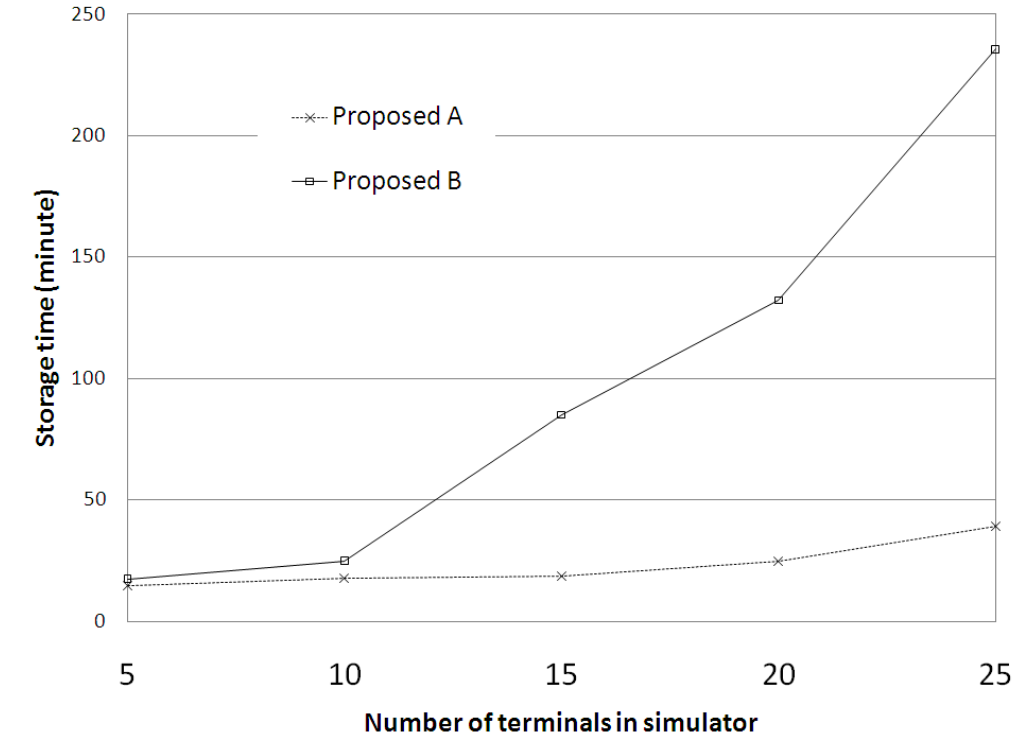


Figure 5.27: Time length of store time (T_{store}).

In order to show the effectiveness of the proposed area control, the following results are shown in detail by focusing on the case where N_{active} is 20. All figures from Figure 5.28 to Figure 5.30 describe the change of each number at elapsed time from the beginning of this experiment. Figure 5.28 shows the number of data-holding terminals in each method. Focusing on the number of passing terminals in the base storage area, N^b_{all}, it starts to decrease after around 8 (minutes) passes. The two numbers of data-holding terminals in the base storage area, N^b_{hold}, in Proposed A and Proposed B follows behind this drop. On the other hand, the number in the extension area, N^e_{hold}, inversely increases.

Next, at around elapsed time 40 (minutes), whereas N^b_{all} and N^b_{hold} increase again, N^e_{hold} decreases in response to these increases. However, after that, N^b_{all} in Proposed A goes down, and finally terminates at around eplapsed time 48 (minutes). With respect to Proposed B, N^e_{hold} drastically grows over N^b_{all}, and firmly maintains its storage capability.

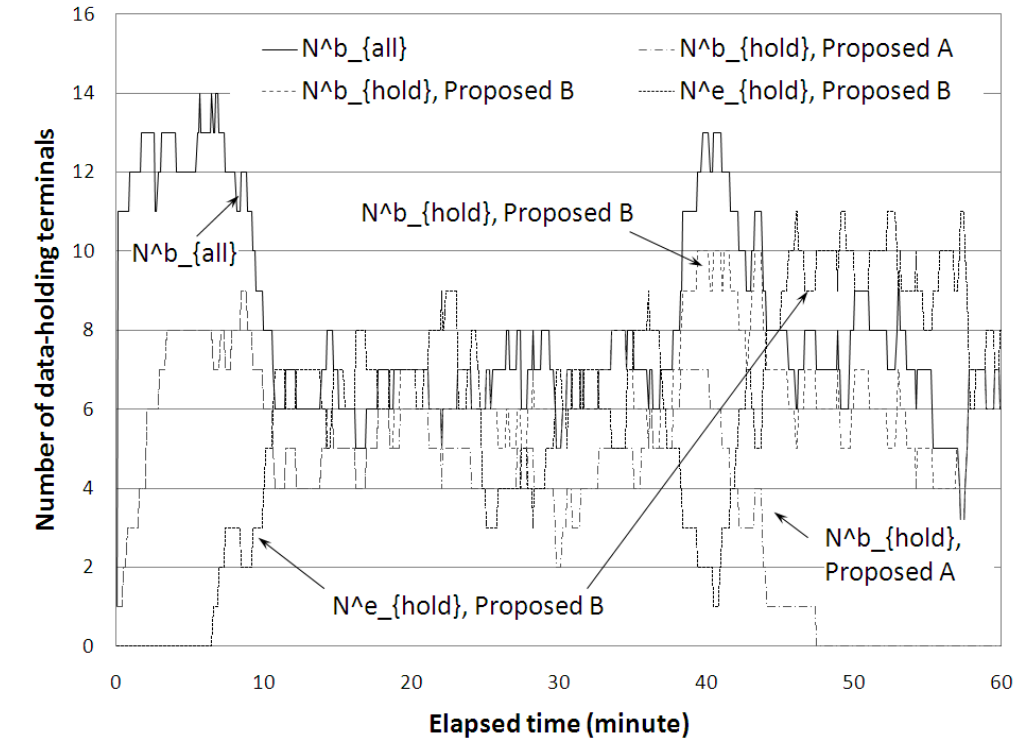


Figure 5.28: Number of Data-holding terminals (N^b_{hold}, N^e_{hold}).

Figure 5.29 also shows the change of the ratio of data-holding terminals, R^b_{hold}. Between elapsed time 20 and 35 (minutes), R^b_{hold} is equals to 1.0 thanks to the help of the storage area terminals. In addition, contrary to the storage capability degradation and termination after elapsed time 38 (minutes) in Proposed A, Proposed B still maintains higher ratios during that period.

Finally, Figure 5.30 shows the number of data-received terminals, N^b_{rece}. N^b_{rece} is the number of terminals that not only pass the base storage area but also receive the store data. It excludes the terminal that receives the data without passing the base area. As Figure 5.30 shows, Proposed B can give higher accumulated numbers than Proposed A.

As a result, the terminals in the storage area can dynamically join the collaborative storage operations when the terminal density gets sparse, and compensate for its degradation of storage capability. We can conclude that the relay area control is a powerful tool to enhance the storage capability.

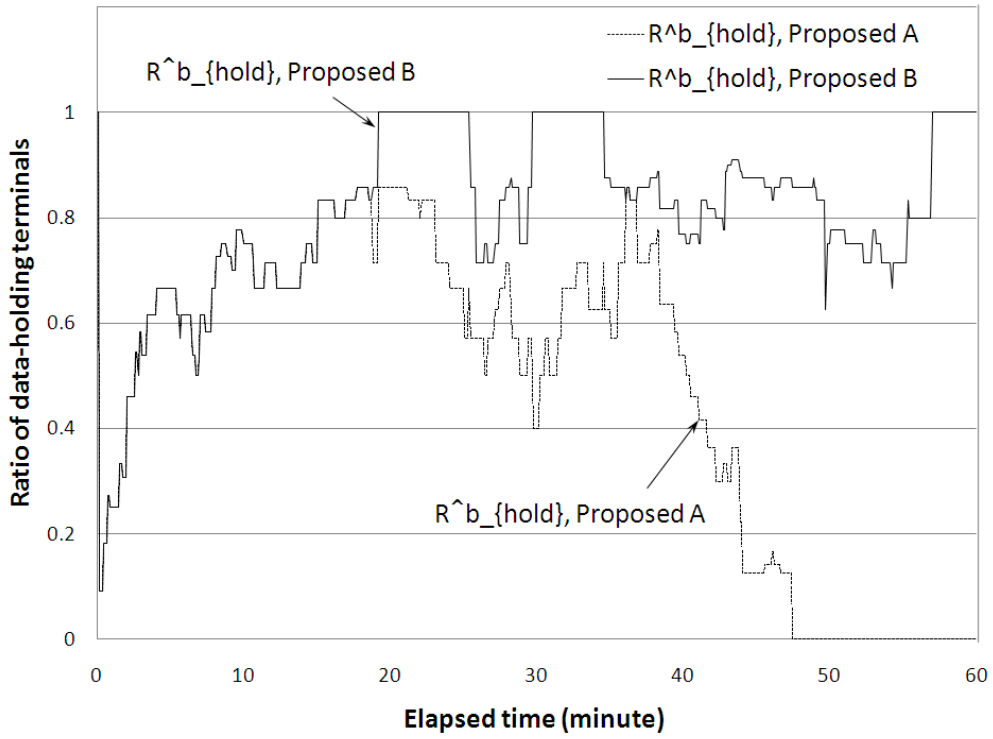


Figure 5.29: Ratio of data-holding terminals (R^b_{hold}).

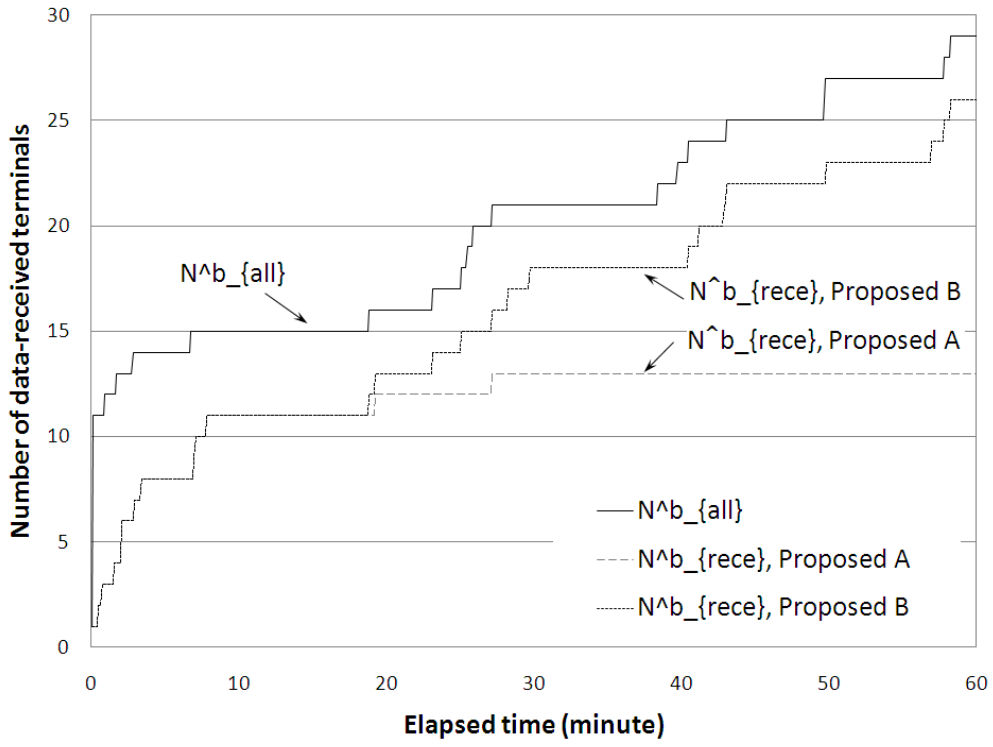


Figure 5.30: Accumulated number of data-received terminals (N^b_{rece}).

The following results are shown in detail by focusing on the case where N_{active} is 50. The elapsed time starts from the beginning of this experiment.

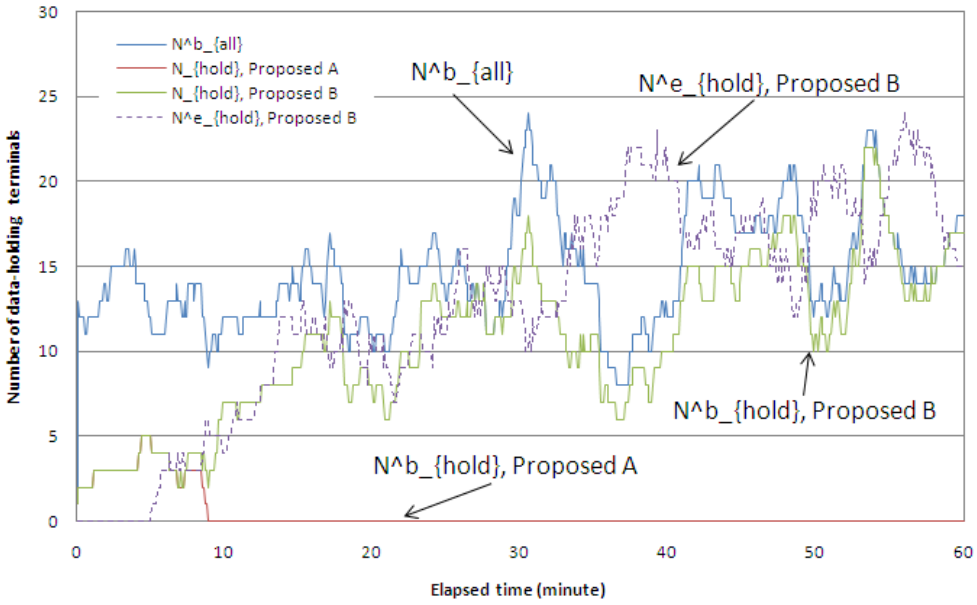


Figure 5.31: Number of Data-holding terminals (N^b_{hold}, N^e_{hold}, N_{active}=50).

Figure 5.31 shows the number of data-holding terminals in each method. Focusing on the number of passing terminals in the base storage area, N^b_{all}, it starts to decrease after around 8 (minutes) passes. The two numbers of data-holding terminals in the base storage area, N^b_{hold}, in Proposed A and Proposed B follows behind this drop. On the other hand, the number in the extension area, N^e_{hold}, inversely increases. Next, at around elapsed time 40 (minutes), whereas N^b_{all} and N^b_{hold} increase again, N^e_{hold} decreases in response to these increases. However, after that, N^b_{all} in Proposed A goes down, and finally terminates at around elapsed time 48 (minutes). With respect to Proposed B, N^e_{hold} drastically grows over N^b_{all}, and firmly maintains its storage capability.

Figure 5.32 also shows the change of the ratio of data-holding terminals, R^b_{hold}. Between elapsed time 20 and 35 (minutes), R^b_{hold} is equals to 1.0 thanks to the help of the storage area terminals. In addition, contrary to the storage capability degradation and termination after elapsed time 38 (minutes) in Proposed A, Proposed B still maintains higher ratios during that period.

Finally, Figure 5.33 shows the number of data-received terminals, N^b_{rece}. N^b_{rece} is the number of terminals that not only pass the base storage area but also receive the store data. It excludes the terminal that receives the data without passing the base area. As Figure 5.30 shows, Proposed B can give higher accumulated numbers than Proposed A.

As a result, the terminals in the storage area can dynamically join the collaborative storage operations when the terminal density gets sparse, and compensate for its degradation of storage capability. We can conclude that the relay area control is a powerful tool to enhance the storage capability.

To analyze the effectiveness of the proposed control in detail, we focus on the specific time. Figure 5.34 shows the number of data-holding terminals in the base area, and expansion area with Proposed A and Proposed B. The number of terminals in the base area decreases around when the elapsed time is 8.5 (minutes) and it means the weakening of the storage capability. After that immediately, the number of data-holding terminals of Proposed A in the base area reach zero. However,

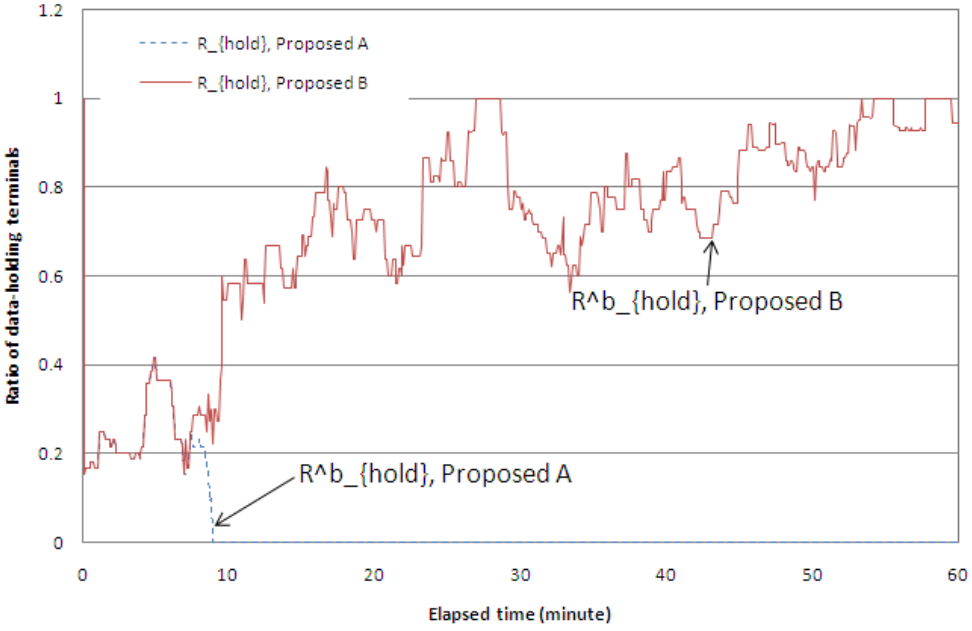


Figure 5.32: Ratio of data-holding terminals (R^b_{hold}, N_{active}=50).

the number of data-holding terminals in the base area of Proposed B keeps around 4 constantly. Therefore, the relay area control save the data to keep in the storage area as long as possible.

Then we evaluate the number of data-holding terminals in the each expansion area. Figure 5.35 shows the number of data-holding terminals in the first, second and third expansion area represented as Storage Area 0, Storage Area 1, Storage Area 2. From the results, the third expansion area Storage Area 2 is not used as expansion area. From these results, there is less data-holding terminals in the third expansion area. Therefore, the number of expansion control is enough to 2 in this situation. The limited number of expansion control depends on the situation, but it can be decided through the same simulation as well.

The number of data-holding terminals while the elapsed time is from 40 to 50 minutes are shown as well. Figure 5.36 and 5.37 show the results of the number of data-holding terminals in the base storage area and expansion area when $N_{active} = 50$. Looking at the term between 35 and 36 minute, the number of terminals and the number of data-holding terminals in the base storage area (N^b_{all}, N^b_{hold}) decrease rapidly. In this point, the storage ability to store the information in the limited area becomes lower. Then looking at the same term in the Figure 5.37, the number of data-holding terminals in expansion storage area (N^e_{hold}), especially in the storage area 2. Therefore, it shows that our algorithm can help to store the information as long as possible when the storage capability of the base storage area decreases.

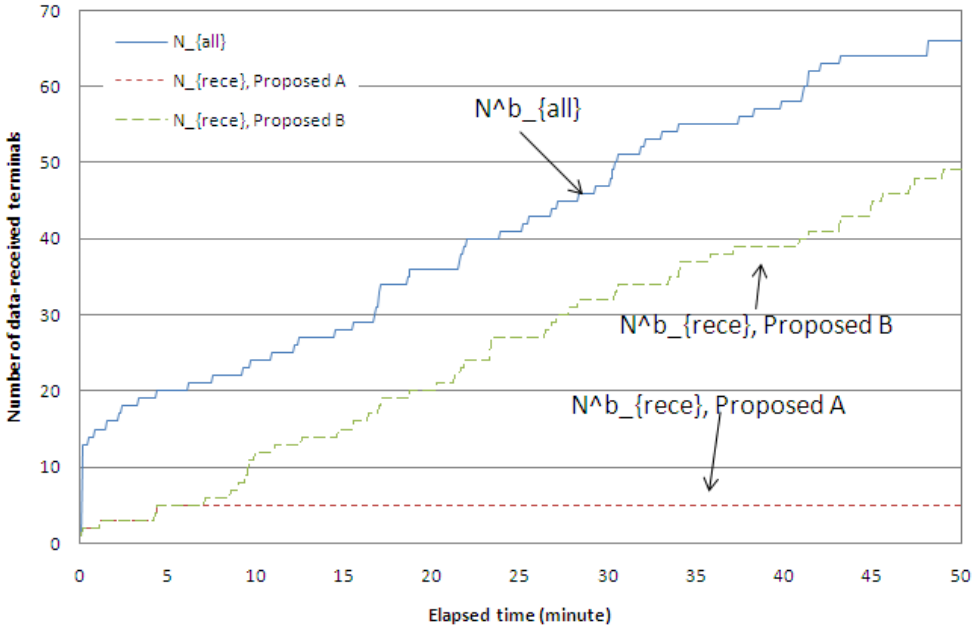


Figure 5.33: Accumulated number of data-received terminals (N^b_{rece}, N_{active}=50).

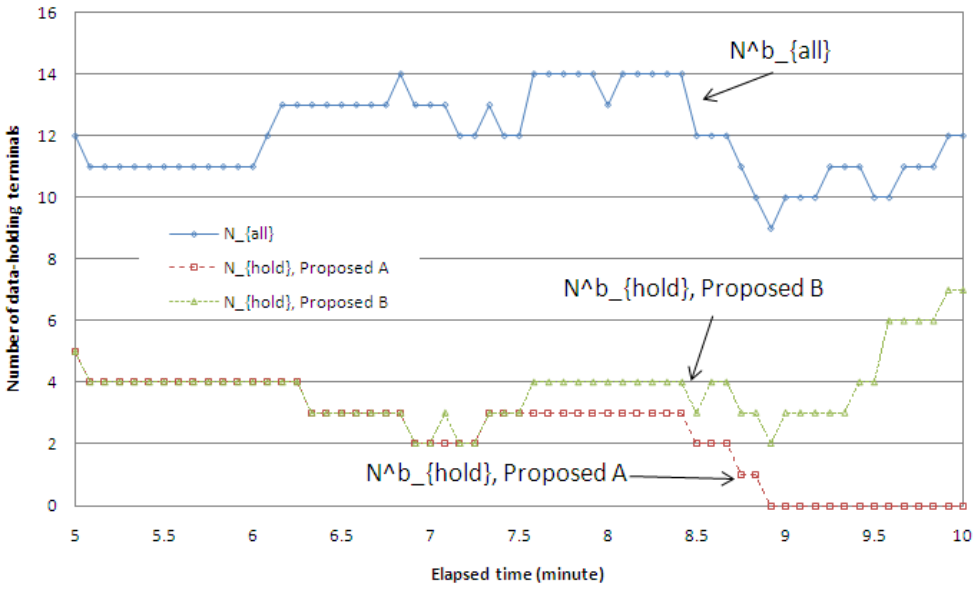


Figure 5.34: Number of data-holding terminals (N^b_{hold}, N_{active}=50).

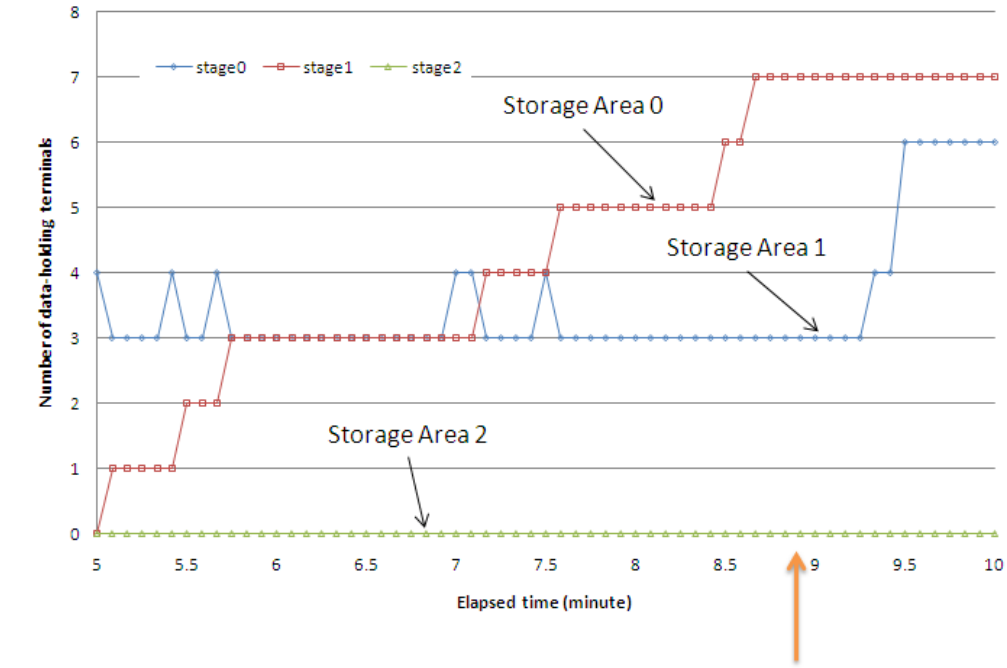


Figure 5.35: Number of data-holding terminals (N^b_{hold}, N^e_{hold}, N_{active}=50).

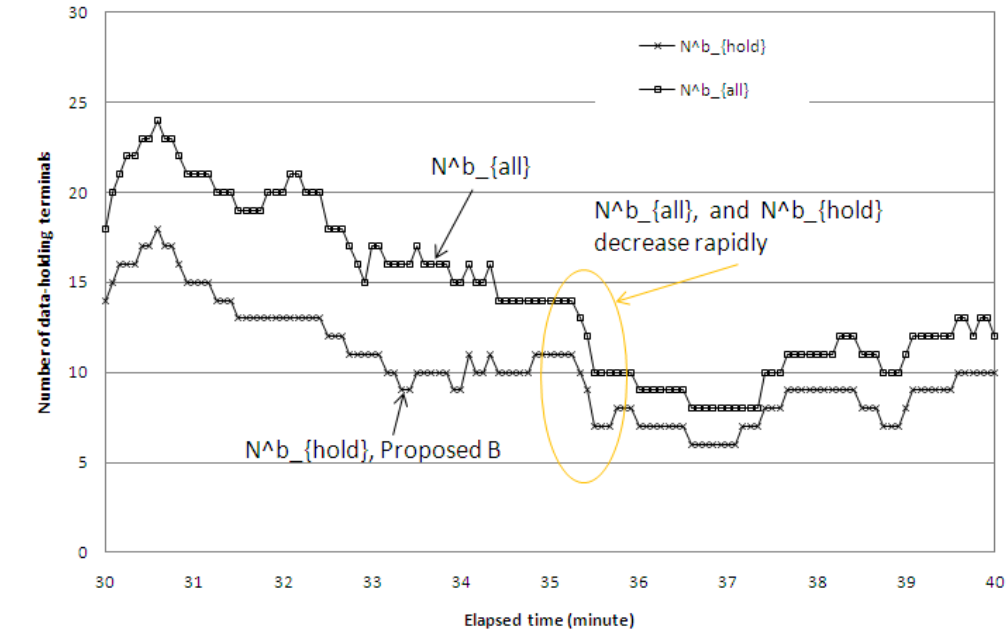


Figure 5.36: Number of data-holding terminals (N^b_{hold}, N_{active}=50).

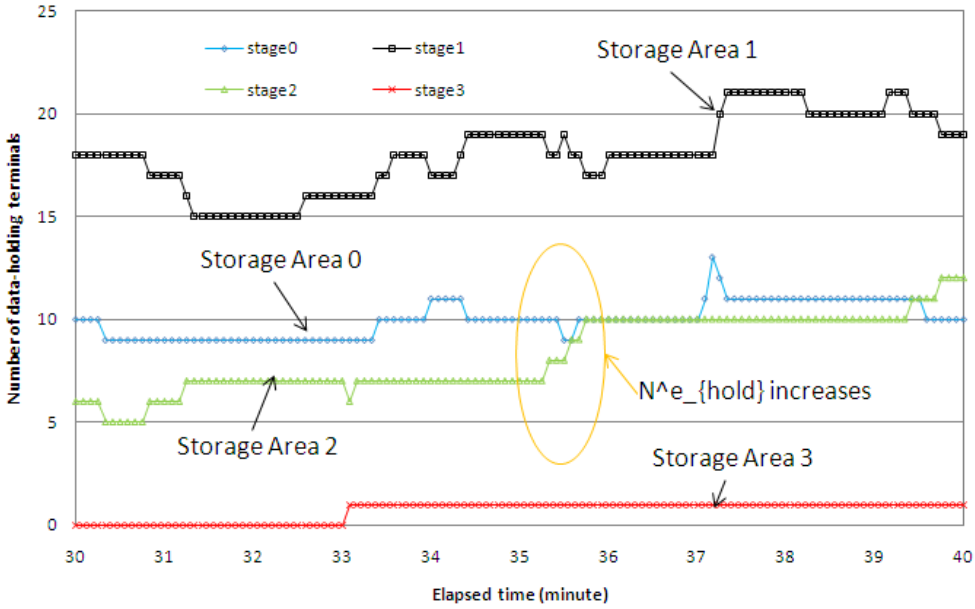


Figure 5.37: Number of data-holding terminals (N^b_{hold}, N^e_{hold}, N_{active}=50, 40to50min).

5.5 Other consideration and application example

5.5.1 Other consideration

Previous Proposed Algorithm

The relay timing and area is decided to store the information within/close to the target area while reducing the number of transmissions between terminals. This behaves as follows; information carrier aggressively relays the information to others in case of fewer terminals around relaying terminal. Meanwhile, more terminals pass nearby, the frequency of the relay is kept lower. The algorithm of deciding relay timing is follows. Once a terminal receives or relays the information, it does not send the information to other terminals until the following (5.4) is satisfied.

$$\frac{Th \times N^b_{hold}}{N_{all}} \leq T_{elapsed} \tag{5.4}$$

where, Th is an control parameter, N^b_{hold} is the number of passed terminals with the target information, N_{all} the total number of passed terminals, and Telapsed indicates the elapsed time since the latest relay timing. Here, Th is determined by the following logic.

- The radius of a general WLAN AP is about from 25 to 30m, and the range of short-range wireless communication of WiFi is about 10m.
- Assuming that the human walking speed is about 0.9m/sec, people take about 22 sec for passing a WiFi-communicable area. When a terminal gets the information, other terminals locating close to the terminal can receive the same information from another terminal. As a result, we can set 22 as the initial Th parameter.

Evaluation Result of Previous Algorithm

We implement our proposed algorithm and the comparable method to a simulator. The simulator configures a people walking model on town streets, and multiple WLAN APs along the streets. People move randomly along the street, the target information is relayed between close terminals. Table 1 shows the simulation parameters. The comparative approaches are follows:

- Baseline 1: The terminal with the information relays it to others inside the predefined communication range.
- Proposed 1: The relay area is limited into the area where one AP of the initial WLAN AP list can be detectable. A terminal does not relay outside that area.
- Proposed 2: The terminal relays the information based on the proposed algorithm. The terminal continues to relay the information until it passes next to the other 3 terminals outside the target area.

In the simulation, the number of WLAN is 9, and of which size is from 20 to 25[m], the whole area is 600[m]x800[m]. An observation time is 10 minutes. The threshold time of passing is 3. Three metrics are evaluated; the storage time, Tstore, which is the time duration that at least one of the terminal relaying the information exists in the limited area, the number of the duplicated transmission to the same terminal Ndup and the number of terminals that have the target information, N^b_{hold}.

Figure 5.38 shows the relationship between the number of terminals in the area, Nterminal, and Tstore, and Figure 5.39 shows the relationship between Nterminal and Ndup. Regarding Tstore, Baseline 1 is as expected, and the Tstore of our proposed method is similar to Baseline 1. On considering Ndup, our proposed method is much less than the others. Therefore, our proposed method can calculate the appropriate timings to store the information in the target area.

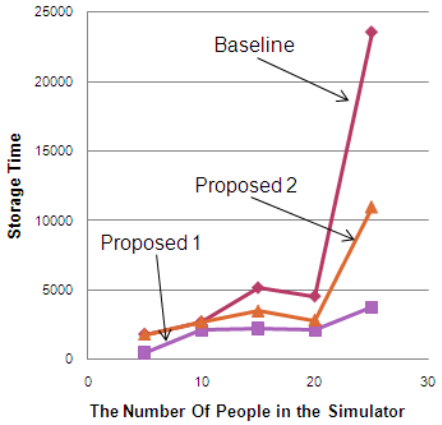


Figure 5.38: Store Time.

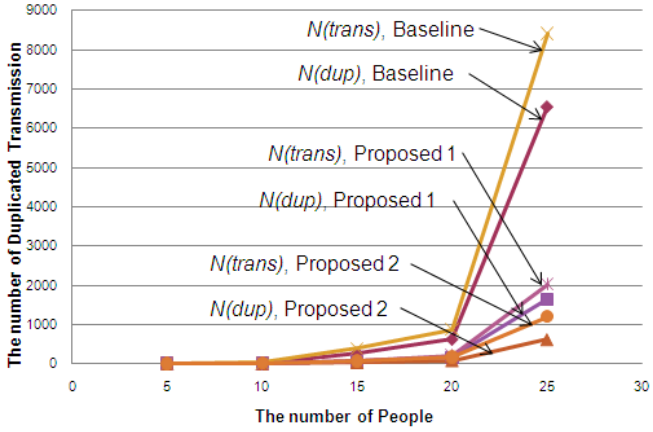


Figure 5.39: Number of relay and duplicative relay.

Figure 5.40 shows the change of N^b_{hold} for each instant of time. While most of terminals in Baseline 1 have the information, Proposed 1 cannot store the information for a long time. However, N^b_{hold} in Proposed 2 is almost constantly a half of the number of terminals in the target area. Considering some terminal are going out or entering the area at each moment, we can say that it should be ideal that a half number of terminals in the target area have the information. Therefore, our proposed method can control the number of terminals that have the information in the target area as constant number and it can store the information similar to the ideal.

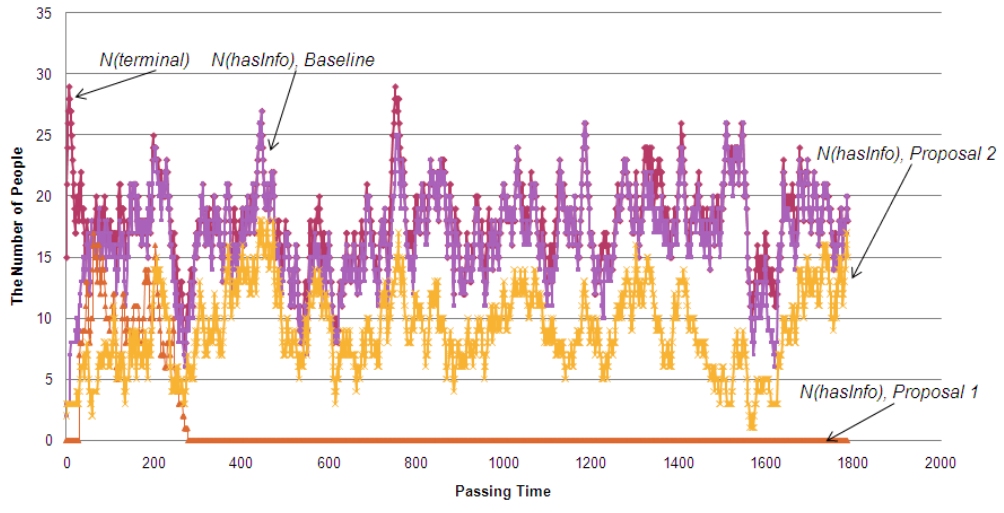


Figure 5.40: Number of data-received terminals.

5.5.2 Application examples

In this section, we describe the application examples using our proposed storage architecture. Figure 5.41 shows an example application to directly appeal some event to the people passing near by. This figure represents the following scenario:

- After the festival, a person thought "this town is too dirty..., so I want to clean!." However, he notices that it is difficult for me to clean all of the area by myself.
- He sends the message which contains the appeal of cleaning town like "Let's clean our town together!" to the person who are surrounding him using portable device.
- Other persons around him receive the notification of the message. The persons can think "Oh, I suppose to that!" by knowing other person's feeling to the town.
- Then, all of the person who sympathize the message can get together and clean the town together.

In this scenario, the purpose of this application is to push the information toward the passing people near the area. There are other application example similar to this application like transmission of rumors.

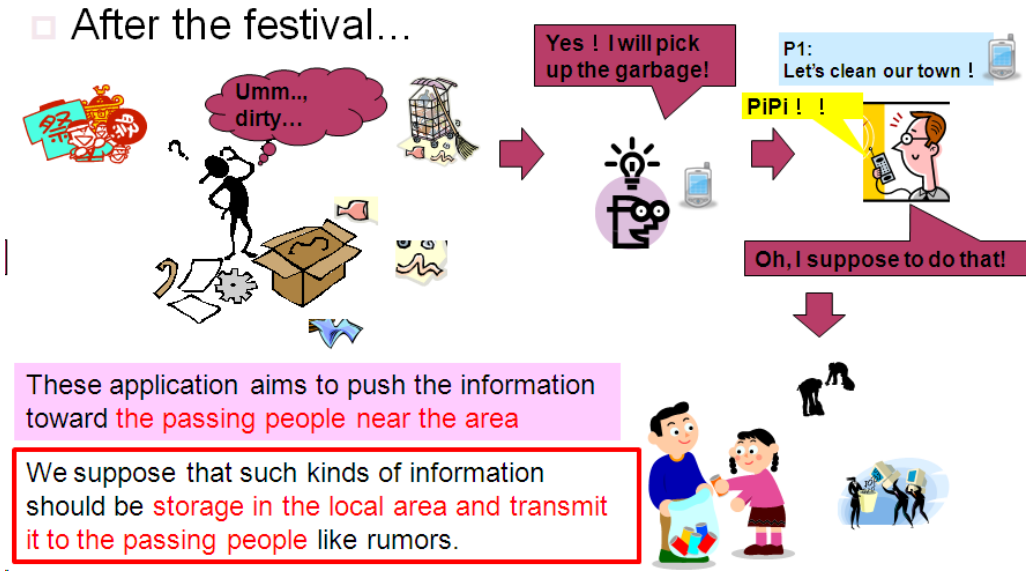


Figure 5.41: Example application (After festival).

Figure 5.42 shows the another example application for entertainment. There is a person playing game using portable phone, and he think "Let's put the item here!", then, the item "sword" is dropped on the game, and the person disappears from the item dropped area. Next time, other person P1 going toward the station passes by the item dropped area. He gets the notification "You can find an item here!" Then, he starts up the game and get the item. On the other hand, another person P2 going toward the station didn't get any notification on his way, because he passed another way to the station. Figure 5.43 shows the example screenshot of the game application. In such application, this item is useful only in the limited area and it's not so important that should be stored on Cloud.

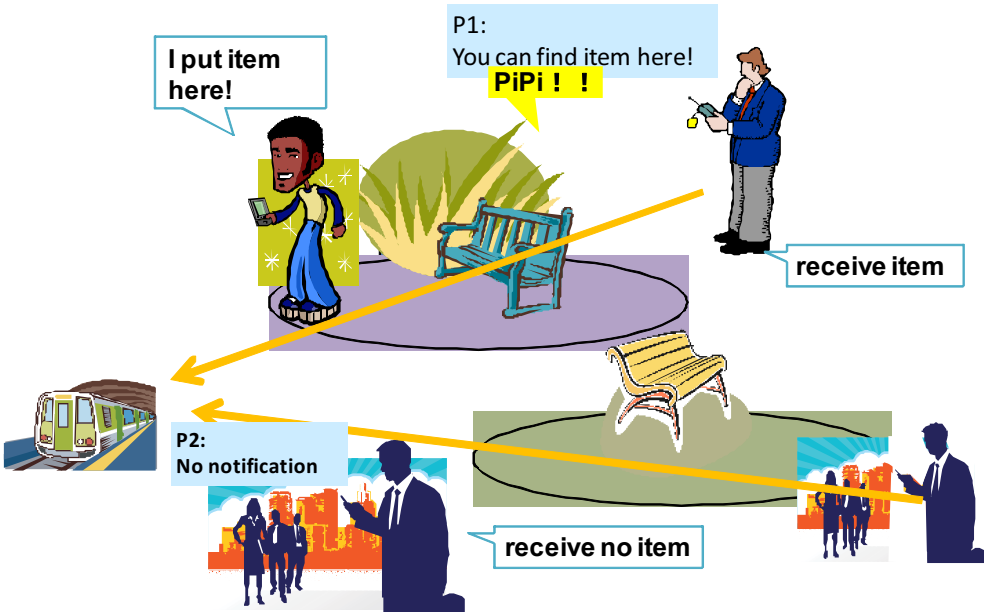


Figure 5.42: Example application (Location based item).

Screenshot when P1 passes through the location.

Screenshot when P2 passes through the location.

Figure 5.43: Example application (Game screen of location based item).

5.6 Summary

This part of the paper tackled a new information storage architecture for location based information. The purpose of the research is to store the such location dependent information in the limited local area without Internet or *Cloud Computing*. It can reduce the amount of data managed on *Cloud Computing*. For this purpose, we proposed a new distributed mobile storage architecture to store information in a local area by collaborating end-terminals. This platform can be installed into each mobile device as a middleware, and can provide temporal storage capability without an expensive infrastructure facility. The proposed storage is maintained by collaboratively sharing and relaying store data among those terminals that pass in/near the local area. Pre-placed storage in the target storage area servers and the Internet infrastructure are not required, and such a storage capability can be provided in anywhere if mobile terminals exit. To achieve such information sharing technology, each terminal cannot know the status of other terminals. Moreover, each terminal which passes the storage area can be the relay terminal, so it is required to reduce the load of relay control. Therefore, we propose an algorithm to control the relay by estimating the number of data-holding terminals in the own communication range.

In addition, the information we treat in this research can be deleted if there is no terminal in the storage area. However, there is a case that no terminal exists in the storage area only temporally. To solve the problem, we proposed the area control algorithm to store the information in the target area as long as possible. By this algorithm, we can achieve the extension of the store time longer than the conventional method.

To show the effectiveness of our algorothm, we evaluated the store time, the number of duplicative relays and the number of data-received terminals. In our research, it is expected that all of the terminals pass through the storage area can receive the information. Threfore, it is important to evaluate the number of data-received terminals. We implemented the simulator that simulates information sharing among terminals that moves in a configured virtual space, and measures storage status and various statics over all mobile terminals. By the experiment, we can show the effectiveness of our algorithm. Our algorithm can relay the information effectively withoug duplicative relay and store the information longer than other conventional method.

The central contribution is that our algorithms achieve lower overhead communication and more robust storage capability.

Currently, we have been developed an embeddable middleware for linux-based embedded operation system, and message exchange application on it. As a future work, we will perform a practical experiment, and show its feasibility and its performance evaluation by using real mobile devices.

Chapter 6

Conclusion

6.1 Summary

In this thesis, the following researches were conducted for the challenging purpose of analyzing and collecting grouped sequences. For the sequential data analysis, we aimed to realize a general analysis method for grouped sequences by converting each raw sequential data to the event sequence. By converting the raw sequential data to an abstracted event sequence, it was possible to extract general requirements for analyzing the grouped sequences. The extracted requirements are the state order, the state duration, the states interval, and the state overlap. We described that we need a model that can deal with these four requirements at the same time and proposed the extended methods step by step based on HSMM. In the first extension, the states interval can be dealt by extending the HSMM that can deal with the state order and the sate duration. We proposed DI-HSMM and IS-HSMM as two possible approaches; a model representing the length of state intervals stochastically and a model representing the states interval as a duration of state. In the first approach, it can model the state interval by introducing the interval length probability in HSMM. There is a possibility to model the negative length probability in the new function. In the second approach, it can model the state interval by introducing the interval state simply. This model settings are simple because it is assumed that non-observation symbol and non-observation state are pre-defined. But, it is difficult to model the state overlap with negative length. Therefore, we further extend the first approach to propose OS-HSMM in order to handle multiple sequences. It can realize the model dealing with the relations between multiple sequences. In each method, the modeling performance and the recognition performance were evaluated and compared with the HSMM. We showed the performance improvement and effectiveness of our model. The extended HSMM makes it possible to model a wide variety of grouped sequences. It is thought that by creating various groups, it will lead to the discovery of new patterns that we did not expect. This realization will lead to the discovery of new patterns.

In aggregating sequential data, we also aimed to virtually realize the aggregation and storage environment only using the terminals including mobile terminals existing the area without using the real device that plays a role of an access point communicating with the sensors and the cloud servers. Our approach is exchanging information between surrounding terminals and mobile terminals, and realizing the temporal virtual storage by borrowing the storage in the terminal. In order to prevent the load due to the number of communication times of the terminals and the disappearance of the information due to the movement condition that no terminal exist in the area, the relay control method and the area control method are proposed. In this relay control, we estimate the possibility that the passed terminal already have the information (have received the information by the other

terminals) and control the timing that the terminal relays the information. In the area control, we estimate the possibility that the information may be lost by estimating the number of terminals existing in the area, and control the size of target area according to the estimated situation. By these two controls, the expected storage was realized with suppressing the load of the terminals within the required storage time given by each application. The simulation results showed the effectiveness of our methods.

Through the research described in this paper, it is possible to efficiently realize the analysis and aggregation of grouped sequences. Since various groups can occur, it is thought that this will lead to the discovery of new patterns that were not supposed. It can be said that it is a new academic contribution.

6.2 Future work

In order to ascertain that this model is used widely in various sequential data, it is necessary to experiment with all possible sequences. It is difficult to examine the effectiveness for various grouped sequences. We will cluster the types of grouped sequences and will examine the representative grouped sequences. In addition to this, it needs too much cost to analyze all assuming grouped patterns of multiple sequences. It is useful that selecting the sequences for the target group. First approach is defining the patterns of grouping by hands, but in the future, it is expected to automatically determine which sequences should be grouped as a target data.

Our approach IS-HSMM assumes that non-observation symbol and non-observation state are pre-defined. If it does not assumed in the practical situation, most complicated training algorithm is needed. We will further extend the model to apply such conditions in the future.

For the aggregation, our proposed approach may lead a new paradigm to realize a temporal virtual storage without allocating access points *a priori*. To apply this approach to the practical fields, it is necessary to consider the security aspect. Since the terminals contributing the service receive and relay the unknown information, it would spread a malware virus if there were malware terminals. To prevent such accidents, it needs the security of the security. It is considered that it is useful to use certificates for reliability satisfaction and keys for encryption and dencryption, or settings that only the information owner can refer its contents uploaded to the cloud server.

Depending on the information or environment design, the terminals may need to analyze and convert the raw sequences into event sequence. Not all the terminals contributing the relay have the application corresponding to the data analysis. Therefore, it is considered that it is useful to attach an identification mark to each sequence in order to judge whether the data is raw or already converted. As well as the identification mark, it is also good to attach the application id to the data. When the sequence is raw data, the terminal can determine whether it analyzes the data by my application by comparing the owned application ids with the application id attached to the data. If it needs to decode the raw data for analysis, it has to be also considered the security aspect.

In order to apply the approach to this environment safely, it needs to consider the security of the security, and some problems may cause newly occur beyond the research field. It needs to consider all the assuming problems and solve them in coordination with other research fields.

References

[1] Clement Atzberger. Advances in remote sensing of agriculture: Context description, existing operational monitoring systems and major information needs. *Remote Sensing*, 5:949–981, 2013.

[2] F Gravenhorst, A Muaremi, J Bardram, A Grunerbl, , O Mayora, G Wurzer, M Frost, V Osmani, B Arnrich, P Lukowicz, and G Troster. Mobile phones as medical devices in mental disorder treatment: an overview. *Journal of Personal and Ubiquitous Computing*, 19(2):335–353, 2015.

[3] H Akaike. Power spectrum estimation through autoregressive model fitting. *Annals of the Institute of Statistical Mathematics*, 21(1):407–419, 1969.

[4] R E Bellman. Dynamic programming. *Princeton University*, 1957.

[5] R E Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME—Journal of Basic Engineering*, pages 35–45, 1960.

[6] G E P Box and G M Jenkins. Time series analysis, forecasting and control. *Oakland, CA: Holden-Day*, 1976.

[7] L A Klimko and P I Nelson. On conditional least squares estimation for stochastic processes. *The Annals of Statistics*, 6(3):629–642, 1978.

[8] Dag Tjstheim. Estimation in nonlinear time series models. *The Annals of Statistics*, 21(2):251–273, 1986.

[9] J Connor, L E Atlas, and D R Martin. Recurrent networks and narma modeling. *The 4th International Conference on Neural Information Processing Systems (NIPS)*, pages 301–308, 1991.

[10] N Golyandina and A Zhigljavsky. Singular spectrum analysis for time series. *Springer Briefs in Statistics*, 2013.

[11] S Kouchaki, S Sanei, EL Arbon, and DJ Dijk. Tensor based singular spectrum analysis for automatic scoring of sleep eeg. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 23(1):1–9, 2014.

[12] H Sakoe and S Chiba. A dynamic programming approach to continuous speech recognition. *Proc. of 7th International Congress on Acoustics (ICA) 1971*, C13, 1971.

[13] H Sakoe and S Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):159–165, 1978.

[14] H Silverman and D.P Morgan. The application of the dynamic programming to connected speech recognition. *IEEE Signal Processing Magazine*, 7(3):7–25, 1990.

[15] E. Hsu, K. Pulli, and J. Popovic. Style translation for human motion. *ACM Transactions on Graphics (TOG)*, 24(3):1082–1089, 2005.

[16] K Kulkarni, G D Evangelidis, J Cech, and R Horaud. Continuous action recognition based on sequence alignment. *International Journal of Computer Vision (IJCV)*, 112(1):90–114, 2015.

[17] S J Julier and J K Uhlmann. New extension of the kalman filter to nonlinear systems. *Proc. of Signal Processing, Sensor Fusion, and Target Recognition (SPIE)*, 3068:182–193, 1997.

[18] B Kusy, A Ledeczi, and X Koutsoukos. Tracking mobile nodes using rf doppler shifts. *Proc. of the 5th international conference on Embedded networked sensor systems (SenSys)*, pages 29–42, 2007.

[19] C Tasse. Nonlinear kalman filters for calibration in radio interferometry. *Proc. of Signal Processing, Sensor Fusion, and Target Recognition (SPIE)*, 3068:182–193, 1997.

[20] G Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[21] L E Baum and T Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966.

[22] C. Churchill. Stochastic models for heterogeneous dna sequences. *Bulletin of Mathematical Biology*, 51:79–94, 1989.

[23] Z Yang and B Rannala. Bayesian phylogenetic inference using dna sequences: a markov chain monte carlo method. *Molecular Biology and Evolution*, 14:717–724, 1997.

[24] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 379–385, 1992.

[25] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Sequential deep learning for human action recognition. *International Workshop on Human Behavior Understanding*, pages 22–39, 2011.

[26] T Vojir, J Matasa, and J Noskovab. Online adaptive hidden markov model for multi-tracker fusion. *Computer Vision and Image Understanding*, 153:109–119, 2016.

[27] S Alqurashi and O Batarfi. A comparison of malware detection techniques based on hidden markov model. *Journal of Information Security*, 7:215–223, 2016.

[28] P K Rallabandi and K C Patidar. A hybrid system of hidden markov models and recurrect neural networks for learning deterministic finite state automata. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 9(11):2318–2325, 2015.

[29] E Zarrouk and Y Benayed. Hybrid svm/hmm model for the arab phonemes recognition. *International Arab Journal of Information Technology*, 13(5):574–582, 2016.

[30] L Lamport. Time clocks and the ordering of events in a distributed system. *Communication of ACM*, 21(7):558–565, 1976.

[31] HT Kung. An optimality theory of concurrency control for databases. *Proceedings of the 1979 ACM SIGMOD international conference on Management of data*, pages 116–126, 1979.

[32] P A Bernstein and N Goodman. An algorithm for concurrency control and recovery in replicated distributed databases. *ACM Transactions on Database Systems (TODS)*, 9(4):596–615, 1984.

[33] P F Tsuchiya. The landmark hierarchy: a new hierarchy for routing in very large networks. *SIGCOMM '88 Symposium proceedings on Communications architectures and protocols*, 18(4):35–42, 1988.

[34] D Estrin, R Govindan, J Heidemann, and J Heidemann. Next century challenges: scalable coordination in sensor networks. *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 263–270, 1999.

[35] G J Pottie and W J Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, 2000.

[36] F Sivrikaya and B Yener. Time synchronization in sensor networks: a survey. *IEEE Network*, 18(4):45–50, 2004.

[37] JN Al-Karaki and A E Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, 2004.

[38] J Yick, B Mukherjee, and D Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, 2008.

[39] J Rezazadeh, M Moradi, and A S Ismail. Mobile wireless sensor networks overview. *International Journal of Computer Communications and Networks*, 2(1):17–21, 2012.

[40] P Karn. Maca-a new channel access protocol for packet radio. *ARRL/CRRL Amateur Radio Computer Networking Conference*, pages 134–140, 1990.

[41] I Chlamtac and M Conti. Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks*, 1(1):13–64, 2003.

[42] V Balakrishnan, V Varadharajan, U K Tupakula, and P Lucs. Trust and recommendations in mobile ad hoc networks. *Third International Conference on Networking and Services (ICNS)*, 2007.

[43] G Xu and Z Yan. A survey on trust evaluation in mobile ad hoc networks. *Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications (MobiMedia)*, pages 140–148, 2016.

[44] V N Vapnik. Statistical learning theory. *John Wiley and Sons, New York*, 1995.

[45] S Abe. Support vector machines for pattern classification. *Springer Science and Business Media*, July 2010.

[46] W J Boscardin and A Gelman. Bayesian regression with parametric models for heteroscedasticity. *Advances in Econometrics*, 11A:87–109, 1996.

[47] D R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society*, 20:215–242, 1958.

[48] L E Baum and J A Egon. An inequality with applications to statistical estimation for probabilistic functions of a markov process and to a model for ecology. *Bulletin of the American Mathematical Society*, 73(3):360–363, 1967.

[49] H Xue and V Govindaraju. Hidden morkov models combining discrete symbols and continuous attributes in handwriting recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(3):458–462, 2006.

[50] Y Bengio and P Frasconi. Input-output HMM for sequence processing. *IEEE Trans. on Neural Networks*, 7(5), 1996.

[51] F Salzenstein, C Collet, S Lecam, and M Hatt. Non-stationary fuzzy markov chain. *Pattern Recognition Letters*, 28(16):2201–2208, 2007.

[52] S Z Yu and H Kobayashi. An efficient forward-backward algorithm for an explicit duration hidden markov model. *IEEE Signal Processing Letters*, 10(1):11–14, 2003.

[53] C D Mitchell and L H Jamieson. Modeling duration in a hidden markov model with the exponential family. *Proc. of 1993 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2:331–334, Apr. 1993.

[54] P Ramesh and J G Wilpon. Modeling state duration in hidden markov models for automatic speech recognition. *Proc. of 1992 IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP)*, 1:381–384, Mar. 1992.

[55] Kevin Patrick Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, Dept. Computer Science, UC Berkeley, 2002.

[56] S Z Yu. Hidden semi-markov models. *Elsevier Artificial Intelligence*, 174(2):215–243, 2010.

[57] K Murphy. Hidden semi-markov models (hsmms). http://www.cs.ubc.ca/ murphyk/Papers/segment.pdf, Nov. 2002. (accessed 2017-01-08).

[58] J D Ferguson. Variable duration models for speech. *Proceedings of the Symposium on the Applications of Hidden Markov Models to Text and Speech*, pages 143–179, 1980.

[59] Y Guédon and C Cocozza Thivent. Explicit state occupancy modelling by hidden semi-markov models: application of derin's scheme. *Computer Speech and Language*, 4(2):167–192, 1990.

[60] J Sansom. Large-scale spatial variability of rainfall through hidden semi-markov models of breakpoint data. *Journal of Geophysical Research*, 104(D24):31631–31643, 1999.

[61] J Sansom and P J Thomson. Fitting hidden semi-markov models to breakpoint rainfall data. *Journal of Applied Probability*, 38A:142–157, 2001.

[62] Y Guédon. Estimating hidden semi-markov chains from discrete sequences. *Journal of Computational and Graphical Statistics*, 12(3):604–639, 2003.

[63] J Bulla. *Application of Hidden Markov Models and Hidden Semi-Markov Models to Financial Time Series*. PhD thesis, Georg-August-University of Gottingen, June 2006.

[64] J Bulla and I Bulla. Stylized facts of financial time series and hidden semi-markov models. *Computational Statistics and Data Analysis*, 51(4):2192–2209, 2006.

[65] M Dong and D He. Hidden semi-markov model-based methodology for multi-sensor equipment health diagnosis and prognosis. *European Journal of Operational Research*, 178(3):858–878, April 2006.

[66] N A Dasu. *Implementation of hidden semi-Markov models*. PhD thesis, University of Nevada, May 2011.

[67] S Z Yu and H Kobayashi. A hidden semi-markov model with missing data and multiple observation sequences for mobility tracking. *Elsevier Science B.V. Signal Processing*, 83(2):235–250, 2003.

[68] M Baratchi, N Meratnia, P J M Havinga, A K Skidmore, and B A K G Toxopeus. A hierarchical hidden semi-markov model for modeling mobility data. *Proc. of 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2014)*, pages 401–412, 2014.

[69] M Esmaeili and F Gabor. Finding sequential patterns from large sequence data. *International Journal of Computer Science Issues (IJSC)*, 7(1):43–46, 2010.

[70] H Shimodaira, K Noma, M Nakai, and S Sagayama. Dynamic time-alignment kernel in support vector machine. *Proc. of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS'01)*, pages 921–928, 2001.

[71] D J Berndt and J Clifford. Using dynamic time warping to find patterns in time series. *Proc. of KDD Workshop*, pages 359–370, 1994.

[72] J M Richer, V Derrien, and J K Hao. A new dynamic programming algorithm for multiple sequence alignment. *Combinatorial Optimization and Applications, Combinatorial Optimization and Applications*, 4616:52–61, 2007.

[73] KR Muller, AJ Smola, G Ratsch, B Scholkopf, and J Kohlmorgen. Using support vector machines for time series prediction. *B. Scholkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, MIT Press, Cambridge*, pages 243–254, 1994.

[74] X Li, M Parizeau, and R Plamondon. Training hidden markov models with multiple observations - a combinatorial method. *IEEE Transactions on PAMI*, PAMI-22(4):371–377, 2000.

[75] P Natarajan and R Nevaia. Coupled hidden semi markov models for activity recognition. *Proc. of the IEEE Workshop on Motion and Video Computing (WMVC 2007)*, 2007.

[76] P Natarajan and R Nevaia. Hierarchical multi-channel hidden semi markov models. *Proc. of the 20th International Joint Conference on Artificial Intelligence (IJCAI 07)*, pages 2562–2567, 2007.

[77] H Narimatsu and H Kasai. Duration and interval hidden markov model for sequential data analysis. *Proc. of International Joint Conference on Neural Networks (IJCNN2015)*, pages 3743–2751, 2015.

[78] S R Eddy. Multiple alignment using hidden markov models. *Proc. of AAAI Third International Conference on Intelligent Systems for Modecular Biology*, 3:114–120, 1995.

[79] H Kobayashi and S Z Yu. Hidden semi-markov models and efficient forward-backward algorithms. *Proc. of 2007 Hawaii and SITA Joint Conference on Information Theory*, 174:41–46, May 2007.

[80] Y He. Extended viterbi algorithm for second-order hidden markov process. *Proc. of the IEEE 9th International Conference on Pattern Recognition*, pages 718–720, 1988.

[81] P Tamarit, J C Cano C T. Calafate, and P Manzoni. Bluefriend: Using bluetooth technology for mobile social networking. *6th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous)*, 2009.

[82] A Mumtaz and F T Shah. Games over bluetooth. *IEEE International Conference on Computer Systems and Applications*, pages 1167–1170, 2006.

[83] L Wang, L Tokarchuk E D Vial, Y Wang, and A Ma. Blue danger: Live action gaming over bluetooth. *6th IEEE International Conference of Consumer Communications and Networking Conference (CCNC)*, 2009.

[84] A S Y Lai and A J Beaumont. Mobile bluetooth-based multi-player game development in ubiquitous computing. *Journal of Computational Information Systems*, 6:4617–4625, 2010.

[85] S Taneja and A Kush. A survey of routing protocols in mobile ad hocnetworks. *International Journal of Innovation, Management and Technology*, 1(3), 2010.

[86] P Padmanabhan, L Gruenwald, A Vallur, and M Atiquzzaman. A survey of data replication techniques for mobile ad hoc network databases. *International Journal on Very Large Data Bases (VLDB)*, 17(5), 2008.

[87] A Boukerche and A Darehshoorzadeh. Opportunistic routing in wireless networks: Models, algorithms, and classifications. *Journal of ACM Computing Surveys (CSUR)*, 47(2), 2014.

[88] M J Pitkanen and J Ott. Redundancy and distributed caching in mobile dtns. *Proc. of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*, 2007.

[89] W Zha, M Ammar, and Ellen Zegura. Proc. of ieee 24th annual joint conference of the ieee computer and communications societies (infocom). *Journal of Computational Information Systems*, pages 1407–1418, 2005.

[90] S Burleigh, A Hooke, L Torgerson, K Fall, V Cerf, B Durst, K Scott, and H Weiss. Delay-tolerant networking: an approach to interplanetary internet. *IEEE Communications Magazine*, 41(6):128–136, 2003.

[91] Y Wang and H Wu. Delay/fault-tolerant mobile sensor network (dft-msn): a new paradigm for pervasive information gathering. *IEEE Transactions on mobile computing*, 6(9):1021–1034, 2007.

[92] V Cerf, S Burleigh, A Hooke, L Torgerson, R Durst, K Scott, K Fall, and H Weiss. Rfc4838: delay tolerant networking architecture. 2007.

[93] H Mineno, Y Kawashima, S Ishihara, and T Mizuno. Collaboration mechanism for mobile ip based link aggregation system. *Journal of Information Processing Society of Japan (IPSJ)*, 47(7):2224–2235, 2006.

[94] G Tsuchida and S Ishihara. Replica arrangement for location dependent data in consideration of network partition in ad hoc networks. *International Journal of Communication Networks and Distributed Systems (IJCNDS)*, 2, 2009.

Publications

Journals

- Hiromi Narimatsu and Hiroyuki Kasai, "State Duration and Interval Modeling in Hidden Semi-Markov Model for Sequential Data Analysis," Springer, Annals of Mathematics and Artificial Intelligence, vol. 81, Issue 3-4, pp.377-403, Dec. 2017. (Section 3)

- Hiromi Narimatsu and Hiroyuki Kasai, "Selective Data Deactivation Mechanism in Sustainable Area-Based Cache for Mobile Social Networks," Springer, Mobile Networks and Applications, vol. 17, no. 6, pp. 820–830, 2012. (Section 5)

- Hiromi Narimatsu, Hiroyuki Kasai, and Ryoichi Shinkuma, "Area-Based Collaborative Distributed Cache System Using Consumer Electronics Mobile Device," IEEE Trans. on Consumer Electronics, vol. 57, no. 2, pp. 564–573, 2011. (Section 5)

Conference

- Hiromi Narimatsu, Hiroyuki Kasai, "Duration and Interval Hidden Markov Model for Sequential Data Analysis," Proc. of International Joint Conference on Neural Networks (IJCNN), p.3743–2751, 2015. (Section 3.3)

- Hiromi Narimatsu, Hiroyuki Kasai, and Ryoichi Shinkuma, "Location-based distributed mobile storage using short-range wireless communication," Proc. of IEEE International Conference on Consumer Electronics (ICCE), pp. 237–238, 2011. (Section 5)

  ※1 ICCE2011 Best Paper Award (Special Category in Storage Technology)
  ※2 IEEE CE Japan Chapter ICCE Young Scientist Paper Award

Domestic Conference

- 成松宏美, 笠井裕之, "複数系列における重複状態を表現可能な隠れセミマルコフモデルの提案," 情報論的学習理論と機械学習研究会信学技報, vol. IBISML2017-49, no. 293, pp.109–114, 2017. (Section 4)

- 成松宏美, 笠井裕之, "隠れセミマルコフモデルを用いた複数系列の重複状態考慮手法の提案," 第19回情報論的学習理論ワークショップ(IBIS), D1–11, 2016. (Poster)

(Section 4)

- 成松宏美, 笠井裕之, “系列間隔状態ノードを有する隠れセミマルコフモデルの評価,” 情報論的学習理論と機械学習研究会信学技報, vol. IBISML2015, no. 84, pp.233–240, 2015. (Section 3.2)

- 成松宏美, 笠井裕之, 新熊亮一, 山口和泰, “Data sharing control method for location-based collaborative mobile storage,” 映像情報メディア学会技術報告, vol. 35, no. 7, pp.15–18, 2011. (Section 5)

- 成松宏美, 笠井裕之, 新熊亮一, 山口和泰, “場所に紐づいた分散ストレージ実現のための近距離無線通信を用いた情報伝搬手法の提案と評価手法の検討,” 電子情報通信学会大会講演論文集, vol. 通信ソサイエティ1，pp. 506, 2010. (Section 5)

Awards

- ICCE2011 Best Paper Award (Special Category in Storage Technology)(※1)

- IEEE CE Japan Chapter ICCE Young Scientist Paper Award(※2)