論文 / 著書情報
Article / Book Information

題目(和文)	ロボット聴覚とプラン認識に基づく環境理解のためのPlan-Intention-Eventフレームワーク
Title(English)	Plan-Intention-Event Framework for Scene Analysis Based on Robot Audition and Plan Recognition
著者(和文)	小島諒介
Author(English)	Ryosuke Kojima
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第10554号, 授与年月日:2017年3月26日, 学位の種別:課程博士, 審査員:井村 順一,天谷 賢治,中島 求,早川 朋久,篠田 浩一,中臺 一博
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第10554号, Conferred date:2017/3/26, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Plan-Intention-Event Framework for Scene Analysis Based on Robot Audition and Plan Recognition

Department of Mechanical and Environmental Informatics
Graduate School of Information Science and Engineering
Tokyo Institute of Technology

Ryosuke Kojima

Abstract

This dissertation addresses scene analysis, which is essential for environmental monitoring and understanding. The main issues for scene analysis are **(Issue 1)** to extract attributes of an event and **(Issue 2)** to reconstruct scene structure from signal data obtained from sensors. The first issue has often been assessed for each attribute; for example, attributes of a sound event can be extracted by different techniques including sound source detection, localization, and identification. The second issue is challenging and only limited research has been reported. More sophisticated analysis requires appropriate integration of these techniques and reconstruction of scene structure in many applications. To fulfill these requirements, we proposed a '*Plan-Intention-Event framework*' (PIE framework) by combining event extraction with plan/intention recognition. This framework provides design guidelines for scene analysis systems by taking **Issues 1 and 2** into consideration. This dissertation also shows the effectiveness of applying the PIE framework to three attractive applications: cooking recognition, bird song analysis, and web session log analysis.

Regarding **(Issue 1)**, this dissertation focuses on how to extract attributes of an event in the presence of sensory noises and obstacles. To solve this problem, the PIE framework provides two approaches: **(1-a)** multi-sensory and **(1-b)** multi-attribute approaches. The first approach **(1-a)** exploits multimodal information, e.g., utilizing multiple audio and visual sensors such as microphones and cameras. To evaluate this approach, an audio-visual multimodal cooking recognition system is introduced in this dissertation. This system extracts audio-visual events utilizing a Convolutional Neural Network (CNN). Experimental results showed the robustness of this system in noisy and/or occluded situations. The second approach **(1-b)** involves the integration of different methods of extracting attributes. This approach can be realized by extending and integrating existing technologies. This dissertation focuses on robot audition technologies, which provide extraction of sound attributes using sound source detec-

tion, localization, separation and identification. In robot audition, these technologies are performed separately, and connected in a cascade manner, which often results in poor performance. To tackle this problem, we propose a new Spatial-Cue-Based Probabilistic Model (SCBPM), which makes it possible to extract these sound attributes more precisely than the conventional methods dealing with attributes separately and their cascade-type integration. The effectiveness of the method is shown in a bird song analysis application.

(Issue 2) can be solved by the reconstruction of a scene by connecting the extracted events to be consistent with back-ground knowledge. Plan/intention recognition is applicable to solve such an issue. However, two issues remain: **(2-a)** acquisition of background knowledge and **(2-b)** incompleteness of the events. Issue **(2-a)**can be solved by obtaining background knowledge from documents on the web. This task is challenging because these documents are not well-structured. The first step is the selection of a cooking domain; that is, a set of recipes on websites, followed by the construction of background knowledge based on a Hierarchical Hidden Markov Model (HHMM), which is often used in probabilistic plan/intention recognition. To show its effectiveness, this construct of background knowledge was applied to multimodal cooking recognition. The latter problem **(2-b)** occurs in realistic situations such as online plan/intention recognition. We propose a new method for plan/intention recognition that can deal with incomplete data using probabilistic context-free grammar (PCFG), a more flexible model than HHMMs. The proposed method introduces two types of computation for a prefix probability for PCFGs, and a plan and intention with the highest likelihood. This method was evaluated by web session log analysis, an application that can improve websites and monitor visitors of websites. The most likely plans were computed from logs of web sessions, and the intentions of website visitors were estimated. We successfully show the superiority of these methods compared with other methods without plan recognition. Chapter 1 introduces the background, motivation, contributions, and organization of the dissertation. Chapter 2 provides a review of literature related to scene analysis. First, research related to visual and auditory processing is described briefly, followed by the description of fundamental techniques in scene analysis, including probabilistic models with SCBPM and plan/intention recognition. Chapter 3 describes the PIE framework and an example of this framework. Chapter 4 explains cooking recognition as an application of the PIE framework. Tasks in this application include **(1-a)** audio-visual event recognition using CNN and **(2-a)** construction of HHMM to represent

information of recipes from websites. Chapter 5 describes an analysis of bird song that can aid ecologists. Robot audition techniques are integrated with the SCBPM **(1-b)**. Chapter 6 addresses plan/intention recognition in relation to web session log analysis. A new method is proposed to deal with incomplete data **(2-b)** in realistic situations. Chapter 7 discusses the applicability of the PIE framework and remaining issues and concludes this dissertation.

Acknowledgements

This work was carried out at Imura, Hayakawa, and Nakadai Laboratory, Graduate School of Information Science and Engineering, Tokyo Institute of Technology. This work has been supported by a lot of people. I sincerely thank to past and present members of Imura, Hayakawa, and Nakadai laboratory.

Primarily, I would like to express my appreciation to Prof. Kazuhiro Nakadai. I thank for his thoughtful supervising my study and for providing me this precious study opportunity. Without his support and widespread advice from technical one to scientific writing and presentation, this work does not exist.

I would like to express my deep and sincere gratitude to Prof. Jun-ichi Imura and Prof. Tomohisa Hayakawa for their invaluable comments for my studies. and for their support for my doctor's course in Tokyo Institute of Technology.

I am indebted to Prof. Taisuke Sato who taught me the fundamentals of artificial intelligence and machine learning. He encouraged me to pursue my study.

I would like to express my gratitude to Prof. Reiji Suzuki in Nagoya University and Dr. Shiho Matsubayashi for comments on my studies and for a lot of their support in my fieldwork.

I am also very grateful to Prof. Charles E. Taylor in UCLA for his suggestions from aspects of ethology and for his support of my stay in UCLA.

I am very grateful to Dr. Osamu Sugiyama for his enthusiastic discussion and technical support in laboratory.

I would like to express my appreciation to all the members of my thesis committee, Prof. Jun-ichi Imura, Prof. Tomohisa Hayakawa, Prof. Koichi Shinoda, Prof. Kenji Amaya , and Prof. Motomu Nakashima for their valuable comments on this thesis.

Finally, my thanks for my family will never be enough.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and motivation

The recent development of storage systems and technologies to manage them has enabled the accumulation of time series data, e.g., sensor and log data, over long periods of time. Because it is hard to read and comprehend these large amounts of data manually, attempts are being made to automatically obtain useful knowledge using a computer. The useful knowledge depends on the situation, but it is common to reveal "something that describes a scene". For example, descriptions of a cooking scene, such as "what he or she is cooking" and "how to cook that", are important as they can lead to cooking support systems and provide advice on cooking procedures. Because such scene descriptions are usually not directly observable from sensory data, it is necessary to estimate the scene descriptions from observations.

Many methods have been proposed to extract scene descriptions from observations. These descriptions can be categorized into the six Ws: *what*, *when*, *where*, *who*, *why*, and *how*. The first four Ws describe the attributes of an event. To answer questions such as "*What*, *where*, and *when* was the event?" and "*Who* performed it?", it is necessary to analyze an event and determine its attributes, including its category, location, and duration. Determine of these attributes of a scene is widely studied for each attribute and sensor. For example, in the field of computer vision, object recognition (what), video event detection (when), region segmentation (where) and facial recognition (who) have been studied. Similarly in the field of audition, sound source identification (what), sound source detection (when), sound source localization (where), and speaker recognition (who) have been studied. Answering questions related to the last two Ws, such as "*Why*

and *how* did you do it", requires a better understanding of an entire scene than of a single event. Information on an entire scene can be extracted by reconstructing scene structure while considering relationship between events, such as co-occurrence or series. Background knowledge, including facts known empirically or derived from pre-analysis, can also aid in the more detailed analysis of an entire scene. Information derived from background knowledge is essential, as it cannot be obtained solely by observation of a scene.

The main problem of scene analysis is much of the information may be unobserved and/or include sensor noises. Especially in real-world applications, factors of unobservability and uncertainty derived from sensor noises should be considered. Unobservability results from limits on the number of sensors and can include dead angles of cameras and limitations of sensing resulting from human intention, which cannot be observed. When a target scene includes unobservable factors, a model-based approach is applicable, which can infer these unobserved factors utilizing a mathematical model. In the field of plan recognition, mathematical models that estimate unobserved factors such as human intentions and plans have been studied. Such models are applicable to scene analysis to estimate scene structure. Plan recognition techniques also enable using background knowledge as constraints on the model. This feature makes it possible to filter out results inconsistent with background knowledge. Sensor noises can also result in uncertainties of estimated information and, even worse, may propagate other errors. In the field of robot audition, especially that focusing on sounds, many techniques to avoid noise-related problems have been proposed. Combining these techniques with other techniques for scene analysis helps reduce the effects of noises.

The topic of this dissertation is scene analysis for real-world applications. Many domain-specific scene analysis systems have been proposed. This dissertation contributes by organizing existing methods from the view point of scene descriptions and by presenting a new framework, the *Plan-Intention-Event (PIE) framework*, which restores aspects found insufficient by the conventional methods. To evaluate the PIE framework, this dissertation also describes three applications: cooking recognition, bird song analysis, and web session log analysis.

1.2 Issues

This section describes issues to be solved in developing an advanced scene analysis framework. This dissertation focuses on the following two issues.

Issue 1. How to extract attributes of an event.

This issue can be formulated as how to extract target information from mixed signals contaminated by noise. This issue has long been discussed in signal processing and other applications. This dissertation discusses this topic in the context of scene analysis.

Issue 2. How to reconstruct scene structure.

Extracting additional information from a scene requires consideration of the entire scene and background knowledge. This issue includes two questions: how to consider the relationship between events and how to exploit background knowledge for scene analysis.

A remaining issue is application-dependent. Solving of this problem requires specific tuning for each application. Because a general framework cannot support application-dependent problems, this dissertation deals with such problems in sections addressing individual applications.

1.3 Our approaches

To deal with the above-mentioned issues, the following approaches have been adopted in the PIE framework.

Approach 1. Integration of information

Our approach to extract attributes of an event from noisy data in scene analysis consists of integration of information. In general, a method requiring integration of multiple types of information performs better than a method dealing with individual pieces of information. For a more concrete discussion, this approach can be divided into two sub-approaches;

(1-a) Multi-sensory approach

A multi-sensory approach, also called sensor fusion, exploits multimodal information, i.e., utilizing multiple audio and visual sensors such as microphones and cameras. This approach is effective when sensors compensate for each other by integrating information. Chapter 4 shows an example of a cooking scene evaluating this approach.

(1-b) Multi-attribute approach

Another approach is the integration of multiple attributes to determine the relationship between them. In scene analysis, attributes of an event are often related to each other. For example the location of an event is a good cue to identify the category of that event. Chapter 5 discusses a model that considers multiple attributes, i.e. the location and category of an event.

Approach 2. Introducing plan and intention recognition

To assess the relationship between events, we propose utilizing plan and intention recognition techniques, in which background knowledge is assumed to be supplied as a plan model. This dissertation distinguishes between intention and plan recognition. Intention recognition is the task of identifying an intention, also called a goal, from events. An intention is often divided into several sub-goals, rather than being achieved at once. Furthermore, each sub-goal may be further divided into several sub-sub-goals. Thus, an intention may be visualized as a tree consisting of the overall goal and its sub-goals (see details in Section 2.3.2). This tree is called a plan, and its determination is called plan recognition. Note that multiple plans can achieve a single intention. By constructing a plan in the manner described above, the relationship between events can be expressed as a relationship addressing an intention and subgoals in the plan. However, two problems must be addressed in applying plan and intention recognition to applications:

(2-a) Acquisition of background knowledge

The first problem is how to obtain background knowledge sufficient for plan recognition, i.e. how to construct a plan model. Knowledge may be acquired from various documents on the web. This task is challenging, however, because these documents are not well-structured. The first step may be the selection of a cooking domain, that is, a set of recipes on websites (Chapter 4).

(2-b) Considering incompleteness of observation

The second problem occurs in a realistic situation, such as an online recognition of plans and intentions. All events required to achieve an intention may not be observed. For example, most future events in online situations are unobservable. Chapter 6 addresses this problem in an application of web session log analysis.

1.4 Organization of this thesis

This dissertation consists of eight chapters (Figure 1.1). Chapter 2 surveys the literature related to scene analysis. The first half of this chapter describes existing work on vision- and audio- based scene analysis, whereas, the latter half explains basic probabilistic modeling that represents relationships, plans and recognition of intention. After describing the literature, Section 2.4 addresses the position of our research relative to previous findings. Chapter 3 explains the PIE framework for scene analysis, considering the two issues described in Section 1.2. The following three chapters describe applications of the PIE framework. Chapter 4 describes a cooking recognition system. Because preliminary results suggested that audio-visual multimodal event extraction could effectively recognize a cooking scene, this chapter describes the multimodal method of resolving the problem of obstacles in a kitchen **(1-a)**. This application addresses construction of a model to represent information about recipes from websites **(2-a)**. Chapter 5 uses analysis of bird song in their natural habitats to evaluate the multi-attribute approach. Although audio data are predominant, these audio data recorded in natural habitats tend to include many extraneous noises, suggesting the possible use of a microphone array and robot audition techniques. This chapter integrates robot audition techniques by introducing a new model for the multi-attribute approach **(1-b)**.

Chapter 6 utilizes web-session log analysis to thoroughly investigate the relationships among events. This chapter deals with more realistic settings than conventional methods. Because such situations give rise to other theoretical and practical problems, this chapter explains these problems and shows their solution **(2-b)**. Chapter 7 provides an insight into the remaining issues and concludes this dissertation.

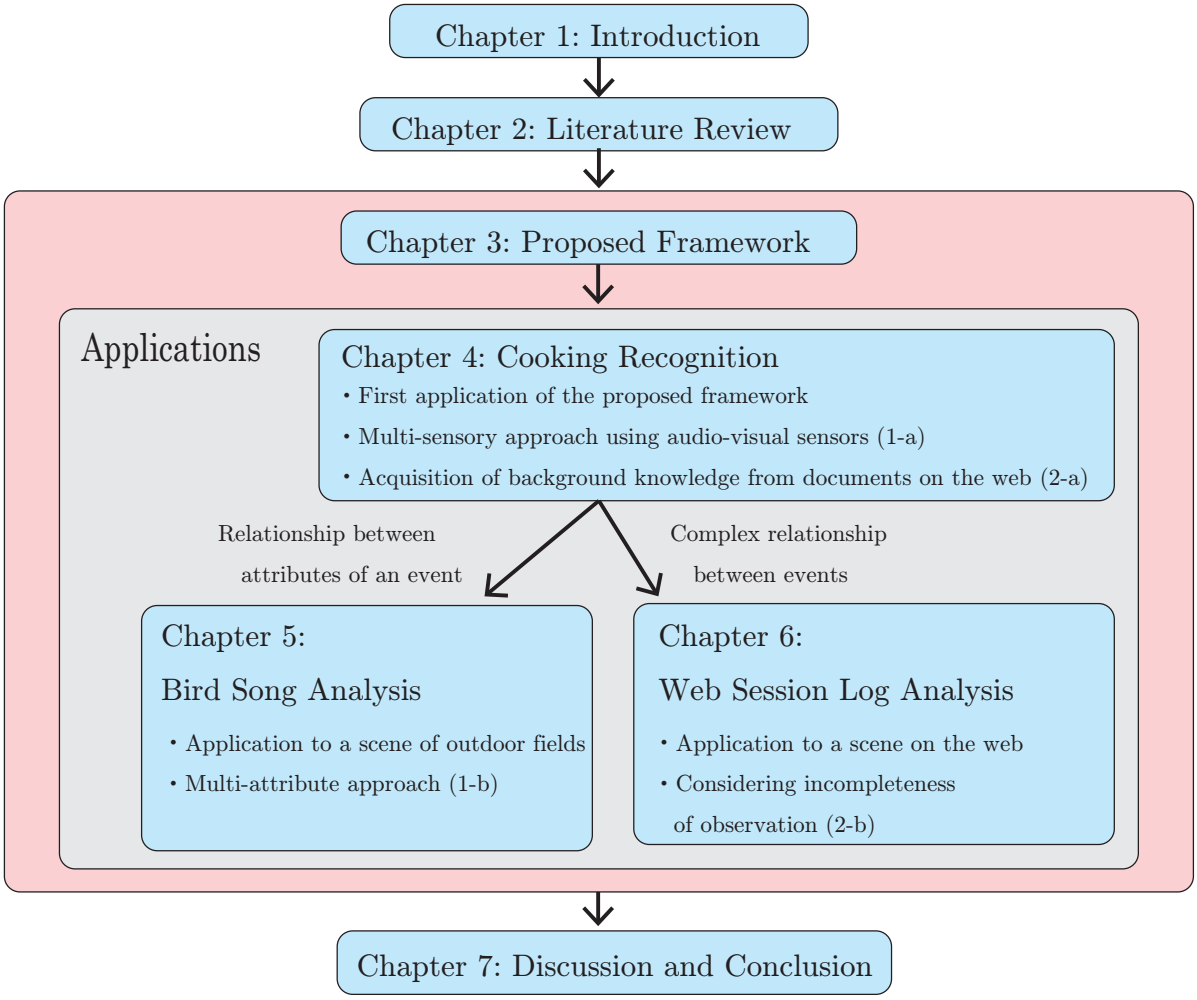


Figure 1.1: Organization of this dissertation

1.5 Refereed publications

This section describes the author's refereed publications relevant to this dissertation. The entire list of publications is shown in the appendix: author's publications.

Cooking recognition and scene analysis framework [Chapter 3, Chapter 4]

- R.Kojima, O.Sugiyama, K.Nakadai: Multimodal scene understanding framework and its application to cooking recognition. *Applied Artificial Intelligence*, 30 (3) pp. 181-200, 2016.

- R.Kojima, O.Sugiyama, K.Nakadai: Audio-visual scene understanding utiliz-

ing text information for a cooking support robot. *International Conference on IEEE/RSJ Intelligent Robots and Systems (IROS) 2015*, Sep., 2015.

- R.Kojima, O.Sugiyama, K.Nakadai: Scene understanding based on sound and text information for a cooking support robot. *Current Approaches in Applied Artificial Intelligence: 28th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE) 2015*, Jun., 2015.

Bird song analysis [Chapter 5]

- R.Kojima, O.Sugiyama, K.Hoshiba, R.Suzuki, K.Nakadai, C. E.Taylor: Bird song scene analysis using a spatial-cue-based probabilistic model *Journal of Robotics and Mechatronics*, Vol.29 No.1, 2017 (Accepted).

- R.Kojima, O.Sugiyama, R.Suzuki, K.Nakadai, C. E.Taylor: Semi-automatic bird song analysis by spatial-cue-based integration of sound source detection, localization, separation, and identification. *International Conference on IEEE/RSJ Intelligent Robots and Systems*, Oct., 2016.

Web session log analysis [Chapter 6, Appendix]

- R.Kojima, T.Sato: Goal and plan recognition via parse trees using prefix and infix probability computation. *Inductive Logic Programming*, Lecture Notes in Computer Science, Vol 9046 pp. 76-91, 2015.

- R.Kojima, T.Sato: A plan recognition method using prefix computation in web access log analysis. *Transactions of the Japanese Society for Artificial Intelligence*, Vol. 29 No. 3 pp. 301-310, 2014 (in Japanese).

- R.Kojima, T.Sato: Goal recognition from incomplete action sequences by probabilistic grammars. *The 24th International Conference on Inductive Logic Programming* (short paper), 2014.

1.6 Notation

The notation used in this dissertation is as follows:

- A boldface capital letter indicates 1) a matrix or 2) a set.

- A non-boldface capital letter indicates 1) a scalar number or 2) a nonterminal symbol (e.g. in context-free grammar).
- A small boldface letter indicates a vector ($\mathbf{x} = (x_0, x_1, \ldots, x_n)$).
- A star indicates an estimation (the estimation of x is x^*).
- $P(\cdot)$ indicates a discrete probabilistic distribution function.
- $p(\cdot)$ indicates a continuous probabilistic distribution function.

Chapter 2

Literature Review

This chapter summarizes the literature on scene analysis and related fields.

Scene analysis comprises research in three areas of research (Figure 2.1): robotics, machine learning and artificial intelligence (AI), and signal processing. This figure shows that scene analysis is in the same research area as robot audition and vision. Scene analysis is a complex technology designed to make a robot or system better understanding a scene using sensors such as cameras and microphones. To review scene analysis and reveal current problems, this chapter is divided into four parts:

- Scene analysis in robot vision
- Scene analysis in robot audition
- Fundamental techniques in scene analysis
- Positioning of this dissertation towards related work

2.1 Scene analysis in robot vision

Conventional scene analysis has been studied primarily in the field of computer vision. Object detection and recognition are popular techniques that extract attributes from an image to answer questions such as "Where and what is the object?". The recent development of deep learning techniques has resulted in great progress in the recognition of general objects [1]. Also Faster R-CNN [2] makes possible high speed computation for both object detection and recognition. Extraction of other attributes from an image

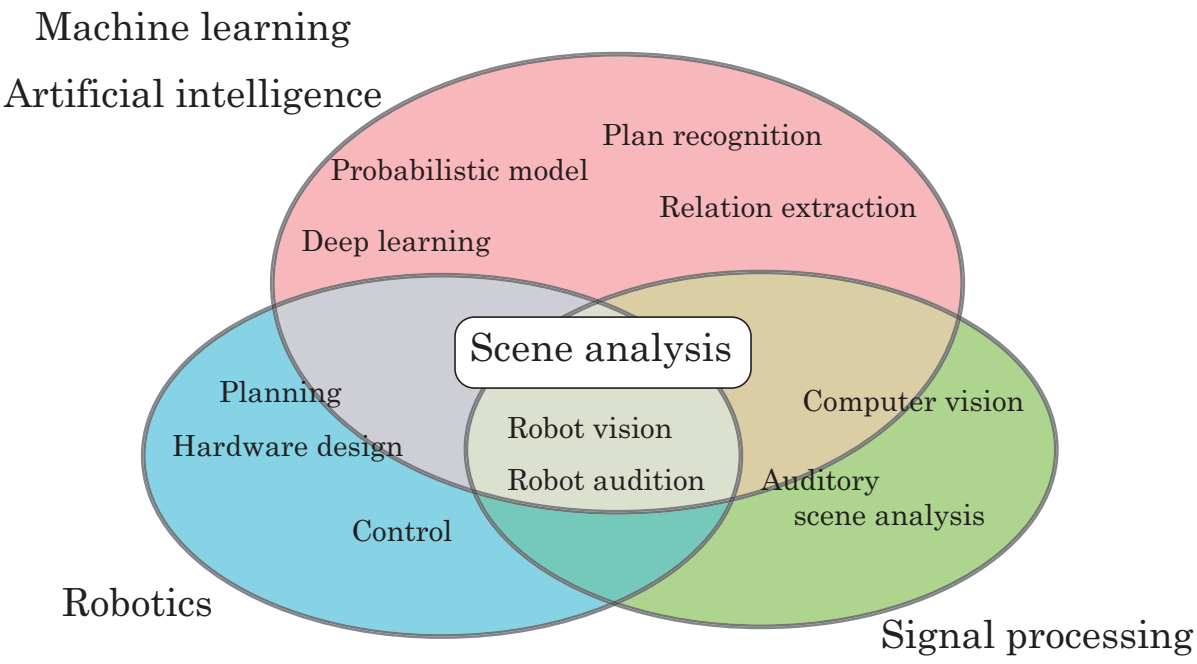


Figure 2.1: Scene analysis

has also been widely studied. For example, extracting descriptive attributes of an image (e.g. "has legs") has been challenged object [3].

Vision-based scene analysis has assessed the relationships between objects and scenes [4]. The method has been introduced to assess activities of groups of persons (Figure 2.2). When we see (A) in Figure 2.2, we cannot distinguish whether the individual is talking to him/herself or chatting with someone else. In contrast, we can easily recognize chatting when we see (B) in Figure 2.2. Thus, *object-object relationships* should be considered. Various methods have been proposed, including co-occurrence and spatial relationships. Another type of relationship is *scene-object relationship*, an approach that utilizes the dependence between an object and a scene. For example, a "car" object is often present in a driving scene, but cannot be present in a cooking scene. A basic method used in both of these approaches is a graphical model, such as Markov networks, in which a relationship between entities is represented as an undirected graph using edges between entities.

In vision-based human activity and video scene recognition, the main focus is time-series information. A detailed review classified approaches to human activity recognition into two categories: single-layer and hierarchical approaches [5]. Single-layer approaches

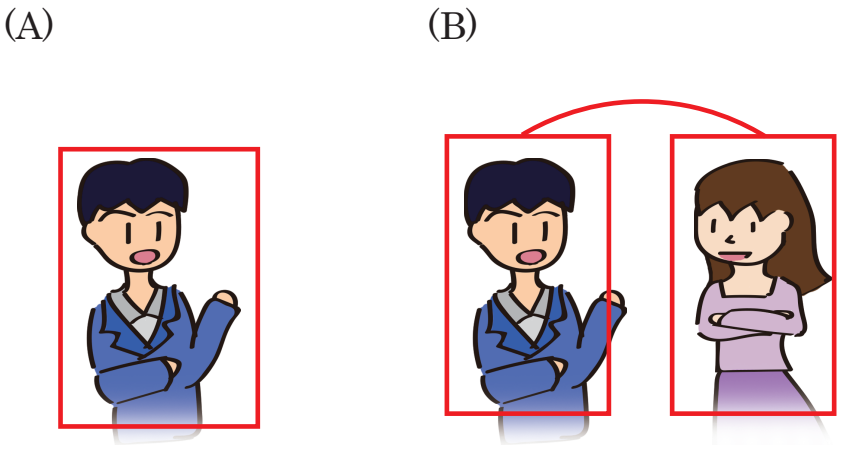


Figure 2.2: Object-object relationships. (A) View of a single person, with no information about whether he is talking to himself or someone else. (B) View of two people, providing additional information that they are chatting.

recognize human activities directly from sequences of images, making them suitable for the recognition of sequential gestures and actions. In contrast, hierarchical approaches represent higher-level activities combinations of simple activities. Recognition systems composed of multiple layers are suitable for the analysis of complex activities. Hierarchical structures can be constructed using a Hidden Markov Model (HMM) and Probabilistic Context-Free Grammar (PCFG), either directly or indirectly. These techniques make possible the representation of the complicated structure of a relationship. Another well-known technique is an AND-OR graph, which can represent the structure of an image or video [6, 7]. Such a model provides structure and algorithms similar to those for PCFGs but focuses on different aspects of modeling.

These techniques can address many challenging tasks in the field of vision. However, much work on vision-based systems has been evaluated in high-visibility spaces or rooms with cameras on the ceiling, thereby avoiding occlusion and dead camera angles. These settings are practical only for surveillance in a building or meeting room; but may not satisfy the conditions in other living environments and low-visibility outdoor spaces. Robots usually use other sensors in such situations, suggesting that this approach should be considered in scene analysis.

2.2 Scene analysis in robot audition

Robot audition is a more recent research field than robot vision. Audition has been investigated to compensate for the drawbacks of vision and to utilize sound information like voices. However, noise robustness is crucial, as sound is always contaminated by noises like environmental sounds. A main issue of robot audition is how to achieve a purpose with noise robustness. Sound source localization, separation, and identification are the most prominent achievements in robot audition. One of the most sophisticated techniques is signal processing using multiple microphones, called a *microphone array*.

Sound sources can be localized by computing a so-called spatial spectrum, representing the power of sound in each direction. These powers can be computed by beamforming approaches, in which a beam is formed for each direction and scanned. Delay and Sum BeamForming (DSBF) for sound is a basic beamforming approach using a microphone array. The transfer from a sound source to a microphone in the array must be computed in advance by simulation or measurement. High resolution and noise robustness in beamforming requires the formation of a sharp beam with respect to the target direction. DSBF utilizes differences in the time of arrival of a sound. MUltiple SIgnal Classification (MUSIC) is another approach to sound source localization that addresses beam shape[8]. It is based on a subspace method, which divides the input space into target signal and noise subspaces using Standard EigenValue Decomposition (SEVD). Several extensions of MUSIC methods proposed to deal with directional noise have shown high localization performances in noisy environments [9]. These approaches use matrix decomposition techniques such as Generalized EigenValue Decomposition (GEVD) [10] and Generalized SingularValue Decomposition (GSVD) [11] rather than the SEVD used in the original MUSIC algorithm. These methods have enabled the successful localization of sounds in noisy environments [12].

Sound source separation is a task by which a target sound is extracted from a mixture of sounds coming from several sound sources. Beamforming methods like DSBF are also applicable to sound source separation. Another separation approach is "Blind Separation," in which several sound sources are separated from observed audio signals based on appropriate assumptions. This type of blind approach requires less prior information and is therefore more suitable to applications of robots and scene analysis. For example, separation methods based on independent component analysis (ICA) assume independence or no higher-order correlation between sound sources [13]. Geometric high-order decorrelation-based source separation (GHDSS) with adaptive step size control is an

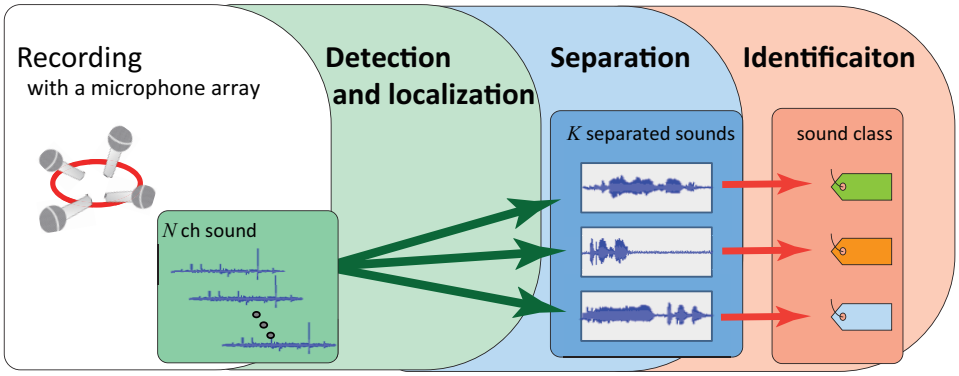


Figure 2.3: Cascaded model for auditory scene analysis

adaptive source separation algorithm [9]. The GHDSS method is an extension of blind source separation (BSS), in which the permutation and scaling problems [14] are relaxed by introducing "geometric constraints" using the transfer function.

Sound source identification is another basic task of robot audition. Identifying a class of sounds is required for a robot to take appropriate actions. This task is highly dependent on problems because target classes and attributes of target signals differ greatly for each problem. A sound source identifier typically consists of two parts: feature extraction and classification. Many features have been proposed for each domain of feature extraction. Mel-Frequency Cepstrum Coefficients (MFCCs) are used in various domains, including speech recognition systems. Other features have been considered for domain-specific tasks, including identifying animal vocalization [15, 16]. Sound source identifiers frequently utilize general classifiers, including Gaussian mixture models (GMMs) and support vector machines (SVMs) [17]. Recent progress in deep learning in speech recognition [18], as well as in competitions on auditory scene classification, such as for example DCASE 2016 [1], have shown that CNN- and DNN- based methods and fusion methods perform well [19, 20]. In such situations, features extracted in an unsupervised manner have been reported effective, including mel energy or spectrograms as classifier inputs. Because these identification techniques were mainly developed for monaural or binaural sounds, sound source separation or the selection of a single channel should be considered preprocessing for identification.

Understanding an auditory scene requires consideration of multiple attributes of a

[1]Detection and Classification of Acoustic Scenes and Events (DCASE) 2016: `http://www.cs.tut.fi/sgn/arg/dcase2016/`

sound. Let us consider an example of extraction locations and categories of sound sources from mixed sounds. A naive system consists of a cascade; that is, it detects and localizes sound sources from recorded sounds using a microphone array, separates the sound sources, and finally identifies each separated sound source (Figure 2.3). For example, the MUSIC method for detection and localization, the beam-forming method for separation, and an identifier using a GMM, based on the acoustic features of separated sounds, is used for identification. This approach has two drawbacks: the propagation of errors of each step and the co-dependency of separated sounds. Error propagation can be tackled by BNP-MAP [21], which estimates the location and separation of sound sources simultaneously by the method with marginal likelihood using a nonparametric Bayes framework. This method works well for steady sound sources. Figure 2.4 shows an example in which the co-dependency of separated sounds should be considered. A separated spectrogram can be easily obtained with a single sound source (Figure 2.4 (A)). However, if there are two sound sources and separation is incomplete (Figure 2.4 (B)), with the inability to completely separate spectrograms, we cannot distinguish whether leakage peaks or not. This phenomenon adversely affects the latter stage, namely, sound source identification. This may be avoided by using other information; e.g., the locations of sound sources. Chapter 5 of this dissertation discusses the co-dependency of separated sounds. There this problem is addressed by a multi-attribute approach **(1-b)**, i.e., utilizing a new model, SCBPM, which considers the locations of sound sources. Utilizing locations of sound sources has also been studied in the field of speech diarization [22]. The main task is detecting human speech in indoor situations, such as meetings. Some studies have combined multiple types of information. For example, utilizing a time delay in the arrival of sound at different channels of a microphone array has been reported to enhance the accuracy of segmentation [23].

Audio-visual multimodal approaches can provide complementary recognition of each event. As described in Section 2.1, it is difficult to understand a scene only from visual information due to occlusion and dead camera angles. This problem may be overcome by multimodality. Scene understanding and related tasks have also been assessed using other sensory data [5]. For example, audio-visual multimodal methods have been proposed for several tasks [24, 25], with these methods showing both noise and occlusion robustness for each task. An application discussed in Chapter 4 also adopts audio-visual multimodal approaches to achieve noise and occlusion-robustness.

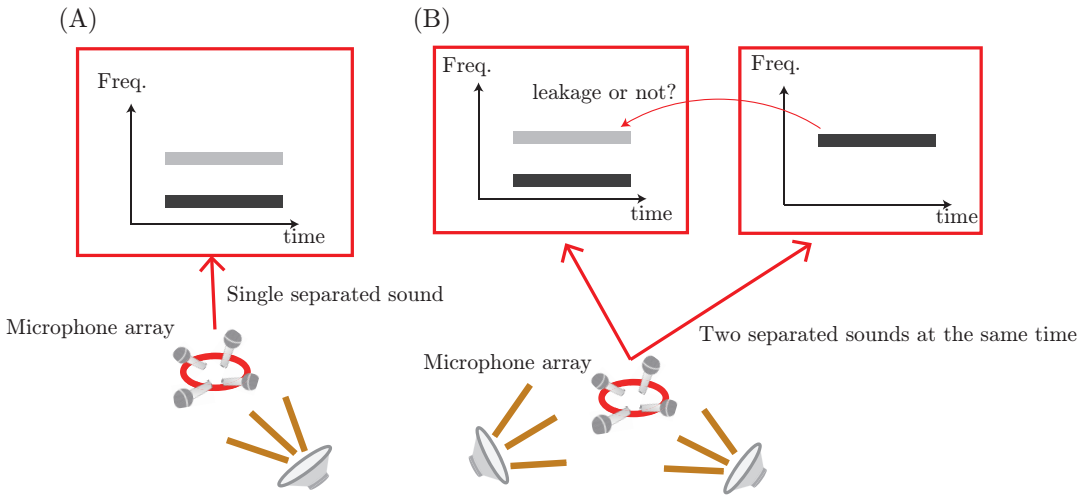


Figure 2.4: Co-dependency of separated sounds: the spectrogram in (A) shows two peaks of a single sound source, whereas the spectrograms in (B) may include a leaked peak due to incomplete separation.

2.3 Fundamental techniques in scene analysis

A main issue of scene analysis is the spatial and temporal integration of extracted information related to a scene. It is essential to consider the relationship and structure of a scene. To model them, this section describes two fundamental techniques: a probabilistic graphical model for relationship and plan and intention recognition.

2.3.1 Probabilistic graphical model for relationship

Probabilistic graphical models provide flexible modeling using graph-based representation. Two types of graphical representations are commonly used: Bayesian networks and Markov networks. Both include factorization and independence related to joint distribution. Bayesian networks use directed acyclic graphs, whereas Markov networks use undirected graphs.

Several models can represent the relationships between two entities. For example, a stochastic block model (SBM) has a simple structure and has been well studied in analyzing social networks, in which relations between two entities can be regarded as a network structure. A simple SBM is a generative model for undirected graphs defined by the following symbols:

- k is the number of groups.

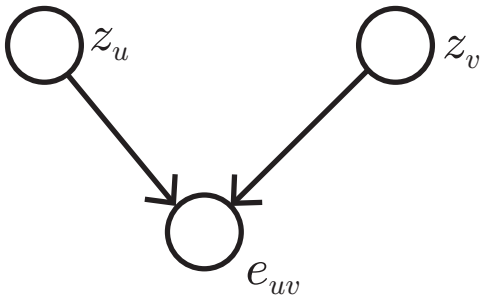


Figure 2.5: Stochastic block model

- u is an index of a vertex.
- z_u is an index of a group of vertices u.
- $M_{ij}(1 \leq i, j \leq k)$ is a probability that a vertex belonging to the group i is connected to a vertex belonging to the group j ($M_{ij} = M_{ji}$).

Each pair of vertices (u, v) is connected at the probability $M_{z_u z_v}$. Therefore, the probability that an undirected graph (V, E) consists of a set of vertices $\mathbf{V}$ and a set of undirected edges $\mathbf{E}$ can be calculated as follows.

$$P(\mathbf{V}, \mathbf{E}) = \prod_{(u,v)\in\mathbf{E}} P(e_{uv} = 1|z_u, z_v) \prod_{(u,v)\notin\mathbf{E}} P(e_{uv} = 0|z_u, z_v) \prod_{u\in\mathbf{V}} P(z_u)$$

where e_{uv} is a probabilistic binary variable representing the presence of an edge between u and v. A Bayesian network representation of this basic structure model is shown in Figure 2.5; in addition, many extended models have been proposed [26, 27]. Similarly, this type of relationship can be represented by a Markov network; although their structures are similar, a Markov network is undirected.

Based on the multi-attribute approach **(1-b)** described in Section 1.3, the proposed SCBPM model described in Chapter 5 can be represented similar to SBM.

2.3.2 Plan and intention recognition

Reconstructing unobservable structures from data is widely studied. Structures related to human intentions are called plans, and the methods determining plans from observations are called plan recognition. Plan recognition, a field of artificial intelligence, has similar problems in scene analysis.

Intention and plan recognition involves determination of intentions and plans based on observations and background knowledge described with logic, rules, graphs, and/or other structures. This task can be regarded as an inverse problem of planning, in which plans and actions are formulated to achieve a given intention. Most basic plan recognition can be addressed by logic approaches such as planning, with the subsequent introduction of probabilistic methods, such as probabilistic latent semantic analysis (PLSA) [28], making it possible to handle the ambiguity of human intentions and plans. Such methods, using a fixed-size Bayesian network, are needed to preprocess and convert raw data into fixed length inputs. This problem may be resolved by use of a grammatical model like PCFGs, in which a plan is regarded as a parse tree and an intention as its top nonterminal symbol. Figure 2.6 shows an example of a plan and an intention for a web shopper. In this case, the only user actions observed are logging onto websites, request for web pages, and buying something on them. On this tree, "Shopping" is the intention and "Searching" is a sub-goal. Among the grammatical approaches proposed are a Bayesian network for actions constructed from a PCFG [29] and its extension to probabilistic state-dependent grammar (PSDG), i.e., PCFG augmented with states [30]. These methods, however, have limitations with respect to CFG rules and depth of recursion; e.g., grammars are not allowed to have left recursion. These limitations are also applicable to other hierarchical methods considered grammatical, such as hierarchical HMM (HHMM) [31, 32] and abstract HMM (AHMM) [33, 34]. Chapter 6 describes a method to relax this limitation, enabling flexible modeling and an application to web session log analysis under realistic settings.

Plan recognition can deal with background knowledge, but acquisition of that knowledge is another problem **(2-a)**. Several methods have been described to collect knowledge from the web and to construct knowledge bases such as YAGO [35] and Google Knowledge Graph [36]. This dissertation describes methods of obtaining such background knowledge from various documents on the web. This task is challenging because these documents are not well-structured. Chapter 4 describes the first step, the construction of a cooking domain; i.e., a set of recipes selected from the web.

2.4 Positioning of this thesis towards related work

Figure 2.7 shows the positioning of this study towards related work with respect to relationships between attributes (**Issue 1**) and the reconstruction of scene structure

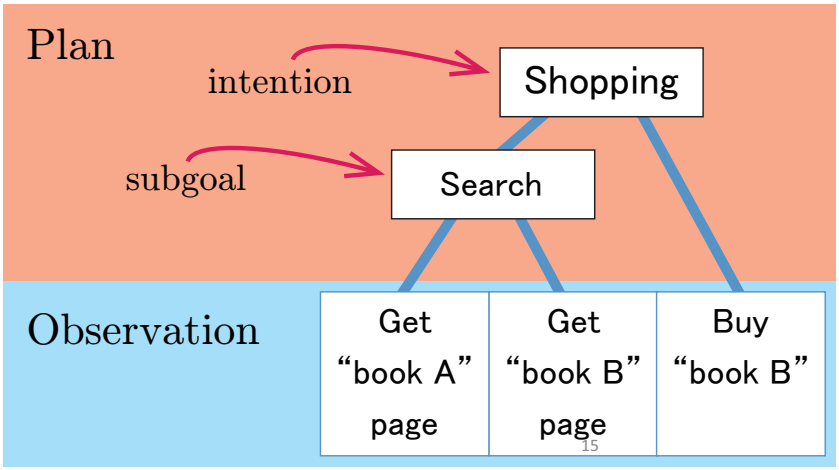


Figure 2.6: Plan and intention recognition and a parse tree

based on the relationship between events (**Issue 2**).

An approach to **Issue 1** in vision is to consider object-object and object-scene relationships (Section 2.1). Also, in audition, the relationship between attributes is considered in several robot audition techniques like the BNP-MAP method [21]. These methods in both vision and audition utilize graphical models to represent such relationships. Probabilistic graphical models are mentioned frequently in the following chapters. Chapter 5 especially describes a new graphical model evaluating the relationship between attributes and its application to bird song analysis. Another approach to **Issue 1** is multi-sensory. As an example, Chapter 4 discusses audio-visual multimodality in cooking recognition.

Issue 2 is addressed in scenario and story recognition for videos by considering the relationship between events. **Issue 2** is also addressed in audition, e.g., speaker dialization. Reasoning techniques to deal with temporal, spatial, and logical structures in video are also studied in plan and intention recognition. Chapter 4 describes the construction of a model using HHMM to represent the relationship between events for a cooking scene. Chapter 6 describes a new method for plan and intention recognition that can address incomplete data using a PCFG, which is a more flexible model, and its application to web session log analysis.

The goal of this dissertation is to complete the PIE framework by filling up the incomplete parts of Figure 2.7. These three applications (Chapter 4-6) have resulted in the extension of scene analysis to more realistic and general cases. Remaining related

Issue1. Extraction of attributes

Approach	Vision	Audio
Basic	Object detection and recognition	Cascade model
Multiple attributes (1-a)	Object-object and object-scene relationship	Not studied well (Section 5)
Multiple sensors (1-b)	Audio-visual multimodal methods (Section 4)	

Issue 2. Reconstruction of scene structure

Approach	Vision	Audio
Basic	Scenario and story recognition	Speaker dialization
	Plan and intention recognition	
Knowledge ackuisition (2-a)	Not studied well (Section 4)	
Incomplete data (2-b)	Not studied well (Section 6)	

Figure 2.7: Positioning towards related work

work of the three applications is introduced in each application chapter.

Chapter 3

Proposed Framework

This chapter describes the PIE framework and its application to cooking recognition, which is also used in the cooking recognition system described in Chapter 4.

3.1 Plan-Intention-Event framework

Figure 3.1 shows the PIE framework, designed as bottom-up processing from sensory data. This framework consists of three layers: plan, intention, and event layers. In the event layer, the attributes of an event are extracted from sensory data using microphones, cameras and other instruments. Typically, such attributes indicate "*When*, *Where*, and *What* was the event" and "*Who* did it". In the intention layer, a label for a given event sequence is estimated utilizing a model that maps from an event sequence to a label. This label is called an intention label, and this model is called an intention model. The plan layer utilizes plan recognition, a technique that determines a plan from an event sequence. This layer determines subgoals for the intention computed in the intention layer and constructs a plan representing the relationship between events through subgoals.

A specialized system is required for each scene, as specification of a system for scene analysis depends on the scene. The PIE framework is expected to promote speedy and efficient development of such systems. The PIE framework provides abstraction of systems, which can determine three functions:

- The PIE framework provides a pattern to design a scene analysis system. This makes it easy to construct the system by instantiating the modules according to the specifications of each scene.

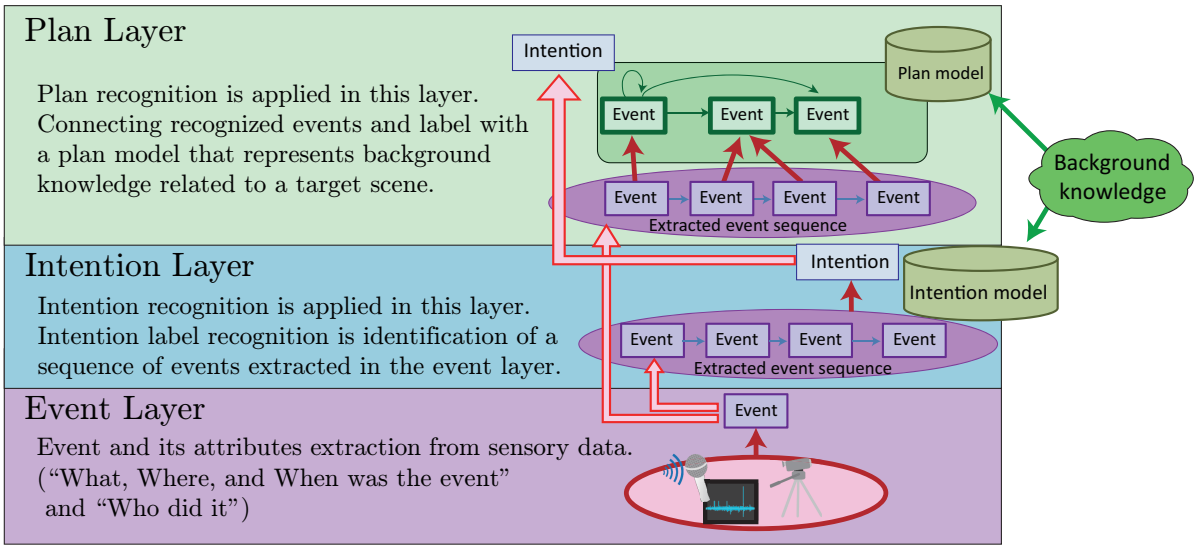


Figure 3.1: The PIE framework for scene analysis

- The PIE framework defines interfaces of layers. Interfaces simplify tasks, thereby, promoting the development of scene analysis systems.

- Layered structure, which each layer outputs readable labels, makes it possible to explicitly utilize background knowledge. When events inconsistent with background knowledge, are determined in the event layers, the plan and intention layers can filter out these events.

This framework provides flexibility to each layer. In the event layer, the types of sensors used and the attributes extracted from data should be considered. System developers can select appropriate algorithms and models for their target applications:

- When multiple kinds of sensors are used, multimodality or sensor fusion should be considered (e.g., Chapter 4).

- When multiple kinds of attributes such as location and time are evaluated, the relationship between them should be considered (e.g., Chapter 5).

In the intention layer, the method of obtaining a label model should be determined in advance. When a large amount of scene data is available, machine learning techniques are effective. In the plan layer, a plan model is required. Although the plan model is assumed to be supplied in manageable form, this assumption may not be satisfied

in many real-world applications, in which background knowledge is usually supplied in unmanageable forms; e.g., a scene description written in a natural language. To solve such problems, the following can be considered:

- When other related data are available, they may be utilized as a constraint on the model. For example, in the cooking recognition system (Chapter 4), recipes are regarded as data related to cooking and are utilized to construct a model.
- Models can also be constructed from data mining results, if the latter are available. For example, in web session log analysis in Chapter 6, pre-analysis involves the clustering of website visitors.
- When the information above is not available, insights may be acquired by visualization step. All the applications described in this dissertation were analyzed by visualization to gain additional insights.
- Another approach is an end-to-end approach, e.g., techniques using deep learning. This approach train a model having many parameters from data. Because this approach requires a large amount of data to train, this approach can be adopted in applications, in which many data are available.

Since such a plan model includes information on an intention label, it is often applicable as a label model in the intention layer.

3.2 Example of the Plan-Intention-Event framework

Because the PIE framework is abstract, we show an example, a cooking recognition system, to aid in understanding the framework. As the details of this system and the cooking recognition application using this system are described in Chapter 4, this section explains that this system is an example of the PIE framework.

Figure 3.2 shows our cooking recognition system. An overview of this system shows that its input is sound and video and its outputs are an event sequence, recipe category, and procedure corresponding to a recipe. Technically, there are three factors for cooking recognition, corresponding to the three layers in the PIE framework. The first was cooking event extraction or the identification of each event, e.g. cutting cabbage, from audio-visual multimodal data. The second was recipe recognition from a sequence of events utilizing a recipe model corresponding to the intention model in Figure 3.1. The

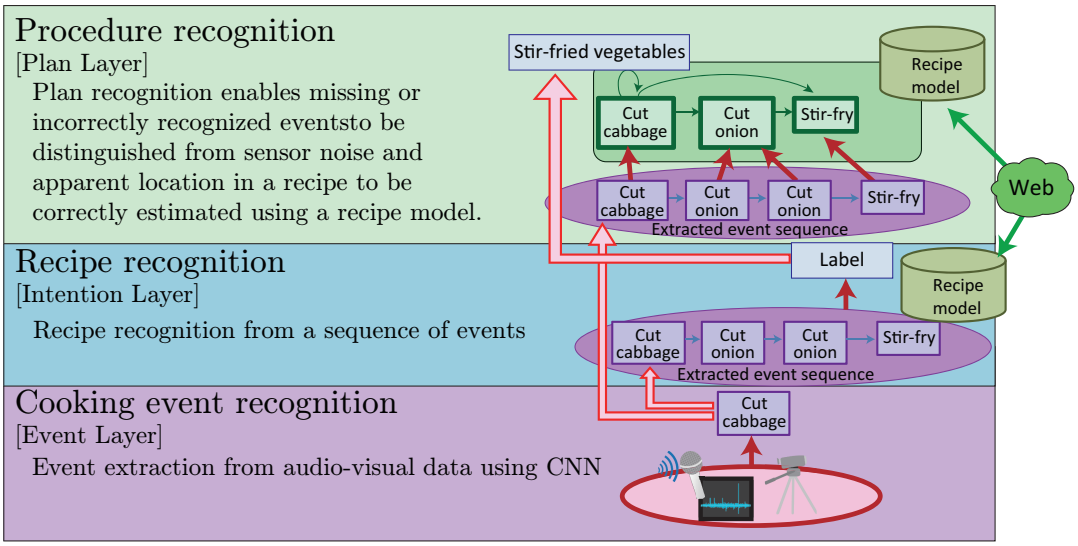


Figure 3.2: Cooking recognition framework

third was procedure recognition; i.e., identifying locations of observed events in a recipe. This model is an application of plan recognition. The recipe model is used again as a plan model in the plan layer.

Key points of this system are audio-visual multimodal event extraction and a recipe model that includes procedures for each recipe category. Chapter 4 shows application of this system with the multimodal identifier and automatic construction of the recipe model.

Chapter 4

Cooking Recognition

This chapter shows an example of the PIE framework and provides solutions to Issue 1, "How to extract attributes of an event", and Issue 2, "How to reconstruct scene structure". The first issue can be resolved by utilizing audio-visual multimodal event recognition. This approach is expected to provide noise- and occlusion- robust event extraction. The second issue in the context of a cooking scene can be interpreted as reconstruction of cooking procedure. A hierarchical hidden Markov model (HHMM), which is frequently used in probabilistic plan recognition, is adapted to model procedures. This chapter also addresses methods of determining HHMM structure and proposes a model utilizing recipes written in natural language.

4.1 Introduction

Cooking recognition is an important technology connected to practical applications, such as cooking support robots. For example, when people cut a cabbage to cook stir-fried vegetables, the robot can provide advices about the next step in the cooking procedure, including the amount of flavoring or duration of boiling. To construct such a robot, recognizing a cooking procedure and utilizing cooking recipes are essential. We call information related to cooking procedures *procedure information*. Although extracting procedure information has been evaluated in fields such as text mining and knowledge discovery [37, 38], less is known about its real-world applications, as in a robot. Our first step in applying the PIE framework to a recipe-based cooking support robot was therefore to extract procedure information.

The event layer in the PIE framework is extracting information related to attributes

of a scene event. A main issue of this layer is "robustness", which depends on sensors. For example, camera-based systems should show occlusion robustness, whereas microphone-based systems should show noise robustness. It is difficult, however, to resolve these two systems. Scene analysis has been mainly studied in the field of computer vision from various viewpoints, such as the attributes of objects [3], their relationships [39, 4] and scenarios [40]. However, scene analysis is difficult to determine from visual information due to occlusion. This problem may be overcome by multimodality. Scene analysis and related tasks have also been assessed using other sensory data [5]. For example, audio-visual multimodal methods have been proposed for several tasks [24, 25], with these methods showing both noise- and occlusion-robustness in each task. Audio-based information also provides good clues for cooking recognition, but some events that depend on visual information, such as the colors of ingredient, are difficult to identify [41]. We therefore, adopted an audio-visual multimodal approach to provide complementary recognition of each event.

Event recognition is not sufficient for scene analysis, as it provides information only on the attributes of individual events. To extract procedure information, we applied the plan and intention layers in the PIE framework. Although these layers are thought to be supplied as a model in manageable form, this assumption is not satisfied in many real-world applications, in which background knowledge is usually supplied in unmanageable forms; e.g., a scene description written in natural language. For example, a cooking scene is supplied as a recipe in natural language. It therefore must be converted to a manageable data structure and both text and audio-visual information must be utilized to extract procedure information.

Since procedure information and background knowledge are highly dependent on an application, a domain should be specified. Furthermore, events exist in many types of relationships, including time-series procedures and co-occurrences. We focused on procedural relationships, since they play important roles in many scenes. We selected a cooking scene as it includes time-series relationships between events, such as cutting cabbages, cutting onions, and stir-frying. In addition, its application is both practical and essential for robots providing cooking support.

To apply the event layer in the PIE framework to a cooking scene, we constructed an audio-visual multimodal event recognizer based on a Convolution Neural Network(CNN), as CNNs have been successfully used in many practical applications, especially in computer vision. We hypothesized that this multimodal CNN approach would provide both

noise and occlusion-robustness in cooking event recognition tasks. We also, constructed a recipe model to utilize recipes considered background knowledge in cooking. We used graphs as internally manageable data structures for recipes. Furthermore, we assumed that cooking procedures could be represented by a hierarchical structure and converted to an HHMM [42], a probabilistic model that can deal with sequential data such as a cooking process. HHMM-based plan recognition was therefore used to extract information.

This chapter is divided into two main parts. In the first, we applied the PIE framework to a cooking scene (see Section 3.2) and present a method to automatically generate a recipe model from websites. Our experiments using real and simulated cooking scene data showed that this audio-visual multimodal approach was superior to single modal approaches and that our cooking recognition system was effective. The second main part presents a case study of our system.

We first introduce the component technologies of our system including the HHMM-based recipe model described in Section 4.3 and the multimodal CNN-based event classifier described in Section 4.4. Section 4.6 shows evaluation of the component technologies through experiments using synthetic and real data. Finally, Section 4.7 shows the application of the prototype to a cooking support system and presents a case study of our system.

4.2 Related work

Cooking recognition is a challenging task and widely studied in scene analysis using various kinds of sensors. Many studies related to cooking recognition utilize visual information obtained from cameras [43, 44]. This approach cannot capture certain actions, however, due to dead camera angles and occlusion. To resolve this problem, multimodal approaches have been proposed. These approaches use an acceleration sensor, a human equipped camera, or other sensors. Sound information can compensate for these drawbacks, as radiated sound can be easily captured in small areas like a kitchen. However, noise robustness is crucial as sound is always contaminated by noise like environmental sounds. This problem can be resolved by using special cooking devices with microphones [45]. These devices, however, are difficult to construct, making it reasonable to use conventional devices like standard microphones and cameras.

Cooking recognition can include three tasks corresponding to the three layers in the

PIE framework. The first is cooking event recognition; i.e., identifying each cooking event such as cutting cabbage, baking, and washing. Most previous research has focused on these basic tasks. The second task is identification of an actual recipe or category of recipes, such as beef stew, or omelet. We call this task cooking recipe recognition. Because this task enables the display of recipe information, some research on applications has focused on this task. The third task is cooking procedure recognition; i.e., identification of one's current location in a recipe. Because this task requires successful recipe recognition, unless a human explicitly indicates a current recipe, this task is more difficult, with little previous research reported. However, it is necessary for a robot to choose an appropriate support by predicting the next step in the recipe. Our goal is to accomplish all of these tasks; to utilize a cooking recipe and procedures involved in the recipe by considering procedure information, defined as the relationship between events, obtained through cooking event recognition.

Various cooking applications have been recently described. For example IBM Chef Watson has made possible the generation of new recipes from existing recipes. Our goal was to utilize recipes for a new cooking support application based on sensor information. Interesting robot cooking applications have been described, including the extraction of cooking procedures from a cooking scene [46]. That study focused on manipulation-level cooking procedures, such as "the right hand grasps a tomato while the left hand grasps a knife to cut it", as the goal of that study was robot manipulation. Recipes were not used, despite recipes being essential for cooking support robots, including messages such as "Next, you should cut cabbages according to a recipe". Another study involved the interpretation of cooking videos. That approach utilized recipe texts and cooking videos with spoken cooking instructions [47] to construct a knowledge base. That study focused on well-maintained video data with low occlusion and scripted texts. In the context of cooking support, spoken instructions are difficult during cooking, whereas it is reasonable to use sounds like those of cutting.

4.3 Construction of a recipe model

This section describes our method of constructing a recipe model represented by HHMM. Our recipe model was automatically generated from cooking recipes on websites. Before describing our method of constructing a model in detail, we briefly describe HHMM.

Table 4.1: Recipe for stir-fried vegetables

Category	Stir-fried vegetable
Ingredients	pork, onion, cabbage, carrot, oil, salt and pepper

procedure	
Step	
Step 1	Cut cabbage, carrots, and onions into bite sized pieces.
Step 2	Put the oil in a wok. Stir fry the garlic over low heat until it is slightly colored.
Step 3	Add the pork and stir-fry over low heat until it is cooked thoroughly.
Step 4	Add the rest of the vegetables and stir-fry for a few minutes. Serve.

4.3.1 Hierarchical Hidden Markov Models

HHMMs are structured multi-level stochastic processes with state transitions (e.g., Figure 4.1). An HHMM H is represented formally by a sextuple $\mathsf{H} = (Q, \delta_h, \delta_v, \Pi, O, F)$ where Q is a set of states, δ_h is a horizontal transition function $Q_d \rightarrow power(Q_d)$, δ_v is a vertical transition function $Q_d \rightarrow power(Q_{d+1})$, Π is a set of probabilities (parameters) of state transitions, F is a set of final states for each level and O is a set of output symbols. Note that $power(Q)$ denotes the power set of Q. The state of an HHMM can be expressed as $q_i^d (d \in \{1, ..., D\})$ and $Q_d = \{q_i^d\}$ where i is the state index and d is the hierarchy level. State transitions of an HHMM can be divided into two types: horizontal and vertical. The probability of a horizontal transition from the i th to the jth state is $P(q_j^d | q_i^d)$ and the probability of a vertical transition from the i th state of the dth level to the j th state of the $d+1$th level is $P(q_j^{d+1} | q_i^d)$ such that $P(q|q_i^d)$ is zero when $q \notin \delta_h(q_i^d)$ and $P(q|q_i^d)$ is zero when $q \notin \delta_v(q_i^d)$. Output symbols depend directly on states of the Dth level alone, and those states, called production states, have additional parameters called output probability $P(o|q_i^D)$, such that $o \in O$. That is, states of the Dth level are the same as those of non-hierarchical HMMs.

In an HHMM, observation is a finite string $\mathbf{o} = (o_1, o_2, ..., o_T)$ of output symbols $o_t \in O$. A joint distribution $P(\mathbf{o}, \mathbf{q})$ is defined as:

$$P(\mathbf{o}, \mathbf{q}) \stackrel{\text{def}}{=} \prod_{(q_{i+1}, q_i) \in Q(\mathbf{q})} P(q_{i+1}|q_i) \prod_{1 \leq k \leq T} P(o_k|q_k^D)$$

such that $\mathbf{q}$ is a finite string of states required to output the symbol string $\mathbf{o}$, and $Q(\mathbf{q})$ is a multi-set of pairs of states (q_i, q_{i+1}), in which $\mathbf{q} = (q_1, ..., q_n), 1 \leq i < n-1, q_n \in F$. A

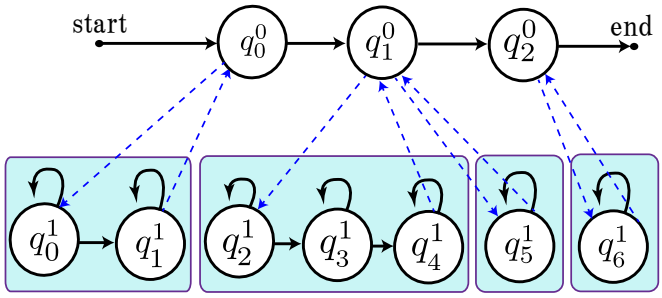


Figure 4.1: Representation of an HHMM

state in Dth level q_k^D is a production state in $\mathbf{q}$ that outputs the kth output symbol. Once the structure (topology) is determined, such as state transitions and a set of parameters Π, the likelihood $P(\mathbf{o}) = \sum_{\mathbf{q}} P(\mathbf{o}, \mathbf{q})$ and the most likely path $\mathbf{q}^* \stackrel{\text{def}}{=} \underset{\mathbf{q}}{\text{argmax}}\, P(\mathbf{o}, \mathbf{q})$ can be computed efficiently using the forward and Viterbi algorithms respectively.

4.3.2 Construction of a flow graph

The first step in constructing a recipe model from a category of recipes is to convert a list of ingredients and a cooking procedure, as shown in Table 4.1, to a *flow graph*, which represents an abstracted recipe and consists of nodes and directed edges (Figure 4.2). Each node shows a cooking event, consisting of a cooking action (e.g. cutting, adding and stir-frying) and its objectives (e.g. onions, cabbages or pork), with each edge showing the order of two events.

A flow graph can be constructed from recipes using a dependency parser by assuming that the order of events is represented by the relationship between words. The dependency parser extracts words representing an action and the corresponding objectives, and constructs a node from those words. The parser also constructs a directed edge according to the dependency relationship between words representing cooking actions. The detailed procedure consists of:

1. Dependency parsing of each sentence in a recipe.

2. Using this result to build nodes consisting of words contained in the action list (action words) and a set of words depending on the action word and contained in the list of ingredients (objective words).

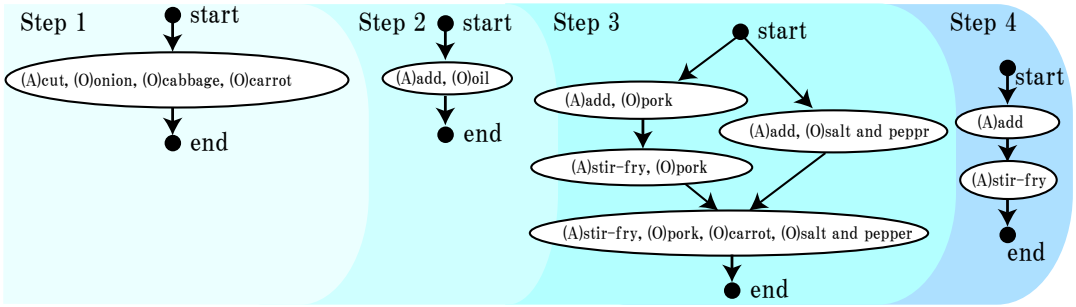


Figure 4.2: Flow graph generated from the recipe in Table 4.1

3. Connecting the nodes using the dependency relationships between their action words.

This procedure converts the recipe in Table 4.1 into the flow graph in Figure 4.2, in which (A) and (O) indicate actions and objectives, respectively. The nodes "start" and "end" are special nodes representing the beginning and end of each step, respectively. For example in step 1, the node "(A) cut (O) onion (O) cabbage (O) carrot" shows a process of cutting onions, cabbages and carrots. A node with two or more parent nodes, like the node in step 1 of Figure 4.3, denotes the non-deterministic order of parent nodes. We call such a node a *joint node*.

4.3.3 Construction of a recipe model

A flow graph was converted to a recipe model, represented by an HHMM, in two steps. We first constructed a non-hierarchical HMM, followed by the construction of a hierarchical structure for an HHMM based on the HMM.

An HMM is a standard probabilistic model for sequential data that includes several parameters regarding state transition and output probabilities. A flow graph can be converted to an HMM through three steps:

1. Each joint node is converted into state transitions in HMM, as described below.

2. A self-loop is added to the HMM for every state.

3. A transition is added from the end node of one step to the start node of the next step.

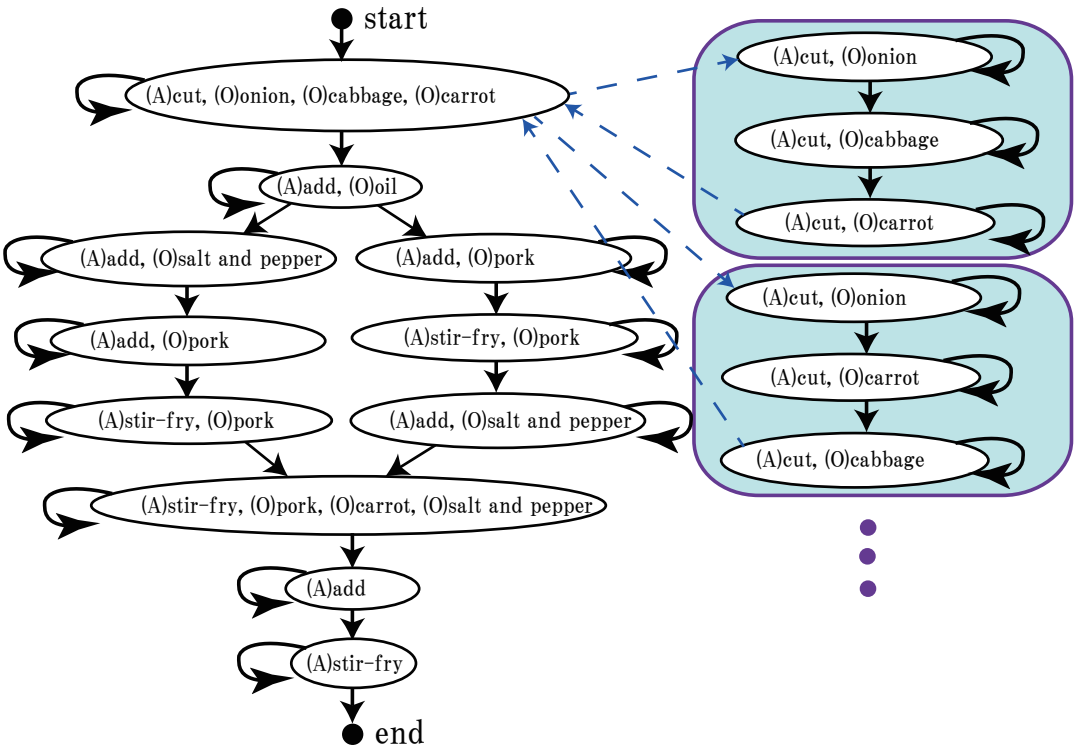


Figure 4.3: Action HMM (left) and event HHMM (left and right)

Each joint node is converted to multiple state transition paths by combining its preceding sub-graphs using an enumeration algorithm (Algorithm 1)[1]. This code contains two functions, parents(*nodes*) and permutations(*set*). The parents(*nodes*) return a set of parent nodes for each given *node*. The permutations(*sets*) return a set of all arrays generated by permutation for each given *set* of nodes. Using these functions, FindPaths($s, t, route, stack$) can compute all possible paths from node s to node t using two stacks: *route* and *stack*. Note the recursive nature of lines 5, 13 and 14, according to three cases. The first case (line 5) is when that path search is finished ($s = t$), but branches remain in the stack ($|stack| > 0$). In this case, the program continues searching from the top branch node in the *stack*. The second case (line 11) is when a focused node t has two or more parents. In this case, after all parents are stacked, one is selected to visit. The third case (line 13) is when a node t should have just one parent node. In this case, the search starts from the parent node. All paths searched are stored in the global variable *result*. All arrays in *result* are converted to state transitions. In this operation,

[1]This algorithm was based on a topological sorting algorithm [48] and algorithms for counting topological orders [49].

Algorithm 1 FindPaths

Require: $s, t, route, stack$

globals $result$
if $s = t$ **then**
  if $|stack| > 0$ **then**
    $c \leftarrow stack$.pop
    FindPaths$(s, c, route, stack)$
  else
    $result$.push$(route)$
  end if
else
  $route$.push(t)
  if $|\text{parents}(t)| > 1$ **then**
    for all seq in permutations(parents(t)) **do**
      FindPaths$(s, seq[0], route, seq[1...] + stack)$
    end for
  else
    FindPaths$(s,$ parents$(t), route, stack)$
  end if
end if
return $result$

the number of states increases exponentially with the number of joint nodes; however this did not cause any trouble in our experiments described in Section 4.6.2 using real recipes.

Application of this procedure to the flow graph in Figure 4.2 yielded the HMM shown in Figure 4.3. The latter was called an "*action HMM*", as each state of the HMM contained only one action and the set of objectives related to it.

A state in an action HMM can be divided into fine-grained states, each of which represents an event consisting of an action and an objective selected from those in the action HMM. An *event HHMM* was therefore defined as an extension of an action HMM, such that each state in the action HMM can be split into finer HMMs of fine-grained states. For example, "(A) cut (O) onion (O) cabbage (O) carrot" can be divided into three parts: "(A) cut (O) onion ", "(A) cut (O) cabbage" and "(A) cut (O) carrot". Permutations of these fine-grained states determines the state transitions in the finer HMM. An example of event HHMMs is shown in Figure 4.3.

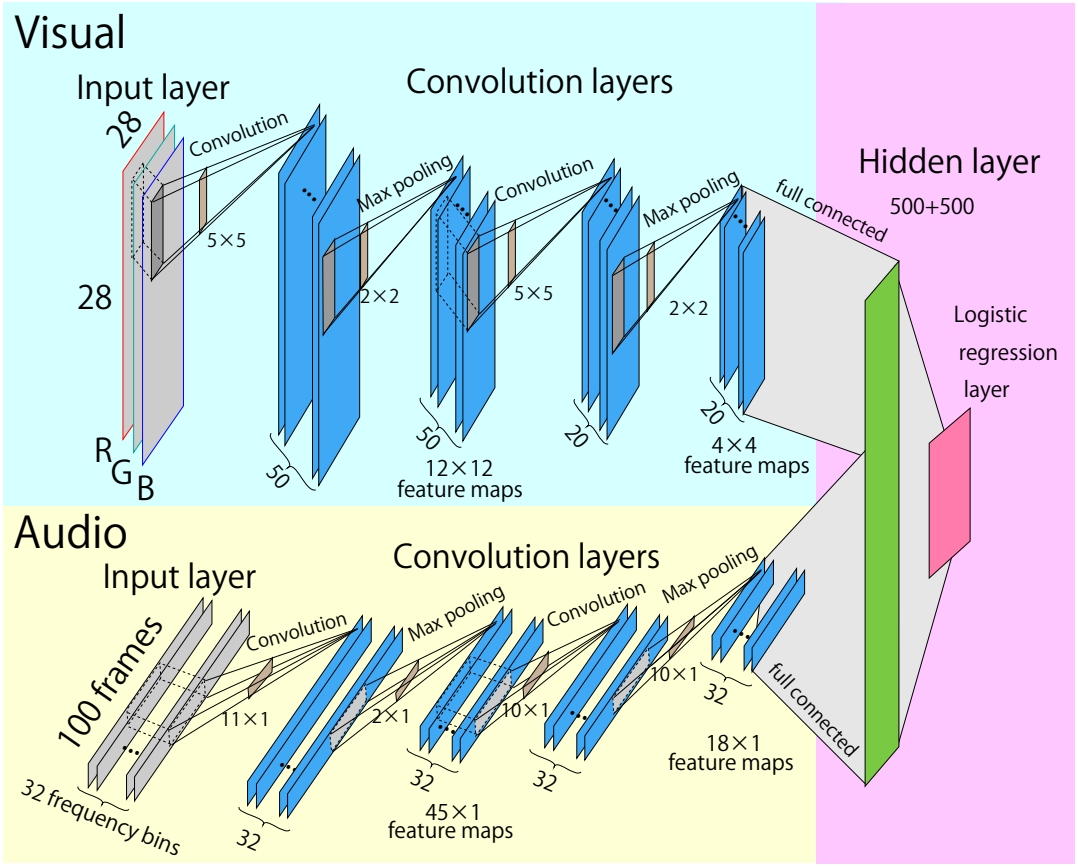


Figure 4.4: CNN structure of an audio-visual multimodal classifier

4.4 Construction of an event classifier

A recipe model provides the most likely cooking recipe and procedure from a given sequence of cooking events by computing the likelihood of a recipe based on an HHMM. To integrate with the recipe model, we also built a CNN-based audio-visual multimodal cooking event classifier, which was trained from preprocessed cooking events and the annotated event labels. Before describing our classifier, we will briefly summarize the concept of CNN.

CNN is a multilayer neural network consisting of input, convolution, hidden, and logistic regression layers. In the convolution layer, response maps are computed with two operations, 1) a convolution operation that convolves the input maps with a number of filters and 2) a max pooling operation that down-samples from the convolved map with a max filter. Classifiers are constructed from hidden and logistic regression layers.

Hidden layers are the same as those of conventional multilayer perceptron; i.e., each operation involves one weight matrix, followed by the application of activation functions to each hidden layer. A logistic regression layer is a regression operator involving one weight matrix and the application of a soft-max function.

CNN can determine the weights of filters at each convolution layer and weight matrix for hidden and logistic regression layers through a learning method involving the minibatch stochastic gradient descent (SGD). Minibatch SGD is a variant of SGD, in which the dataset is partitioned into m mini-batches and parameters are updated with a mini-batch per iteration in SGD.

Using the CNN, we built an audio-visual multimodal cooking event classifier (Figure 4.4). This network structure consisted of separate audio and video parts and a shared part to deal with these two types of inputs. These input layers correspond with input data. RGB data (28×28 pixels) were used as visual input data and a log-scale spectrogram with 32 frequency bins as audio input data(single channel) [2]. The separated parts containing two pairs of convolution and max pooling layers work as conversion for each dataset. RGB data were extracted from recorded video each second. The visual part consisted of two 2-D convolution layers, the first having 20 filters of size 5×5 and the second having 50 filters of size 5×5. Recorded sound was converted to a frame consisting of 32 frequency bins per 10 ms and a concatenate of 100 frames. The input was an audio matrix(32×100) recorded each second. The audio part consisted of two 1-D convolution layers, the first having 32 convolution filters of size $11(\times 1)$, and the second having 32 convolution filters of size $10(\times 1)$, with the shared part containing one hidden layer and one logistic layer to fuse audio-visual information. The hidden layer concatenates audio and visual data and includes 1000 nodes (500 nodes for each part). The output layer of this network is logistic regression and outputs one category from 11 categories of events.

4.5 Cooking recognition system

Figure 3.2 shows the outline of our proposed system; the three layers illustrated can be realized by the recipe model and the event classifier. Cooking event recognition is performed using an audio-visual multimodal event classifier based on CNN, with param-

[2]Because these parameters (especially dimensions of input matrices) are trade-offs between computational time and accuracy, we chose these parameters as computational times of about one second on a PC with Intel Xeon 2.90GHz and Tesla K20c GPU.

eters learned from annotated video with sound. Its output is cooking events, such as cutting cabbage. Cooking recipe recognition is performed by computation of likelihood. The HHMM-based recipe model, described in Section 4.3.3, converted from a flow graph, described in Section 4.3.2), via an HMM, enabling the likelihood of each cooking recipe to be computed efficiently. The cooking recipe with the highest likelihood, as well as recipes with the top-n likelihoods, could be computed, with the latter being important for interactive applications (mentioned in Section 4.7). The most likely hidden states of HHMM, which can easily be computed by the Viterbi algorithm, can be regarded as a cooking procedure in our model. Thus, cooking procedure recognition involves identifying the most likely hidden states of an HHMM-based recipe model.

Differences between recipes and actual cooking must be considered to utilize this recipe model, and cooking events must be recognized from actual cooking. During actual cooking, people often do nothing or perform redundant motions between events. These aspects are omitted from recipes and are therefore not included in the recipe model constructed above. To compensate, we added deterministic self-loop edges for no actions to our model. These edges restrict state transition to another state when no action are observed.

Parameters of HHMMs are usually learned from data. However, we determined them ad hoc, as it is difficult to collect a sufficient number of recorded cooking sounds. This is a future task. The transition probabilities were determined as uniform. The output probabilities of the same action and objective as those associated with the state were determined to be 0.99. The output probabilities of the other symbols were determined to be 0.01.

To implement HHMM, we used a logic-based programming language intended for symbolic-statistical modeling of PRISM [50].

4.6 Experiments

Our cooking recognition system was analyzed using three experimental settings. The first preliminary experiment evaluated an event classifier and determined cooking event recognition by our system. In the second experiment, we used artificial data to evaluate the noise-robustness of a recipe model, as determined by recipe estimation. In the third experiment, we assessed cooking procedure recognition. For the second and third experiments, we used data from 11 categories of recipes (Table 4.2). These categories

Table 4.2: Recipe categories and number of recipes in each

Category	No. of recipes	Category	No. of recipes
Beef stew	32	Ginger pork saute	35
Macaroni au gratin	16	Okonomiyaki	26
Plain omelet	16	Udon	25
Stir-fried vegetables	94		

were collected since this site yielded a sufficient number of recipes for each category and variances in cooking procedures for these recipes were small. Using keywords, we searched for and downloaded recipes belonging to these categories from COOKPAD [3] , in which the recipes are written in Japanese [4]. Table 4.2 shows the number of recipes in each category.

This system was evaluated by recording the cooking events shown in Figure 4.8. Three sets of difficulties were encountered in recording these data. The first was occlusion; e.g. Figure 4.8 (2) shows the occlusion of a left hand,an occlusion not observed in Figure 4.8 (1). The second difficulty was sound similarity. For example, similar sounds are generated when of cutting carrots and radishes, making it difficult to distinguish these steps based on sounds alone [41]. Because the pictures in Figure 4.8(3) and (4) are clearly different, however, cutting carrots and radishes could be distinguished using our multimodal approach. The third difficulty was the similarity of visual information. As a camera was positioned near the cutting board to observe cutting events, the cooking stove could not be observed by the camera. Thus, the visual information of no action and baking events was almost the same scene as in (5) and (6), but their sounds differed. Using this dataset, we evaluated our system by assessing these difficulties. Note that the same kitchen and devices were used throughout these experiments.

4.6.1 Experiment 1 : evaluation of the event model

In this experiment, we addressed the task of cooking event classification to evaluate our classifier. We compared our multimodal classifier with visual-only and audio-only classifiers. The visual-only classifier consisted of a visual part, a hidden layer and a logistic regression layer (Figure 4.4). We prepared 11 classes of cooking events, as shown in Table 4.3. Here, we regarded one second as one frame consisting of a video frame

[3]COOKPAD: `http://cookpad.com/`

[4]We used a Japanese morphological analyzer Mecab [51] and a Japanese dependency structure analyzer CaboCha [52] to build a recipe model.

Table 4.3: Dataset (number of events)

cut radishes	715	cut green onions	207
cut Chinese cabbage	186	cut carrots	391
cut green peppers	290	cut cabbage	195
cut onions	538	cut pork	221
wash	132	bake	2300
no action	846		

Table 4.4: CNN-based cooking event recognition

	audio	visual	audio+visual
Accuracy(average)	82.03	95.13	96.76
Standard deviation	1.36	0.45	0.54

measuring 28×28 pixels and 100 audio frames.

Table 4.4 shows accuracy, as evaluated by 5-fold cross-validation using each classifier, for 6,021 events in the dataset. The audio-visual multimodal classifier was more accurate than the other classifiers, with all differences in accuracy being statistically significant by t-tests (p=0.05). Detailed evaluation of the results showed that "bake" events were difficult to classify using the visual-only classifier, because the camera could not visualize the stove. The precision of classification of "bake" events, which was 73% visually, increased to 90% using our multimodal approach. Recall also increased from 82% to 95%. These results showed that our multimodal approach was effective, even under conditions of a dead angle of a camera and occlusion.

4.6.2 Experiment 2 : evaluation of the recipe model

This experiment evaluated recipe recognition using synthetic data generated by event HHMM, in which the transition probabilities were uniformly distributed. The generated data can be regarded as ideal; i.e., the accuracy of cooking event classification was 100%. To consider more realistic scenarios, we added noise to each event in an event sequence at a probability r, with r being event recognition error rate. That is, noise can randomly replace one event by another event, corresponding to an erroneous recognition of cooking events. We assumed that the estimation succeeds when the recipe used for generation is the same as the estimated recipe. We defined recipe recognition (top-1)

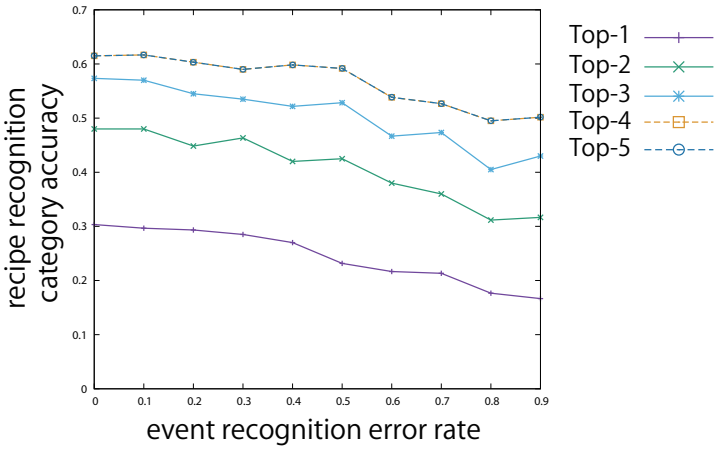


Figure 4.5: Accuracy of recipe recognition category using the event HHMM at artificial event recognition error rates

category accuracy Acc as

$$Acc = \frac{N_t}{\#\ \text{of generated data},}$$

where N_t is the number of estimated recipes containing the same category as the recipes used to generate data. By controlling r, we determined categorical accuracy. We also defined top-n *category accuracy* as the ratio of top-n estimated recipes including the correct category, to all recipes.

Figure 4.5 shows a result of this experiment, during which ten sequences were sampled from one recipe and ten recipes were evaluated for each category. That is, 100 synthetic event sequences were evaluated for each category. Section 4.6.1, the accuracy of cooking event recognition is 96%, which corresponds to a 0.04 error rate. Hence we guess that this system works around 0.04 in this graph where the system performs relatively high accuracy. Results showed that consideration of the top-n categories can prevent noise-associated reductions in category accuracy (error of event classifier). This is very important, as the classification error was higher in this than in the previous experiment due to variable real world noises (also described in the next experiment). Thus, a system, in which the top-n categories are presented to the user and the user selects an appropriate recipe, is practical to use in real world situations.

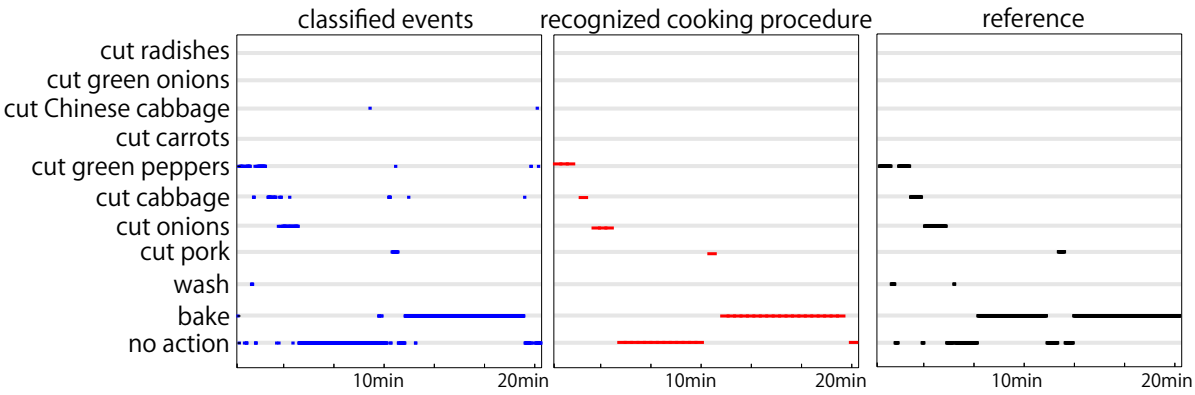


Figure 4.6: Output of the event classifier (left), result of procedure recognition (center), and reference (right)

4.6.3 Experiment 3 : cooking procedure recognition

In the third experiment, we extracted a cooking procedure from a real cooking scene, in which dishes belonging to a category, stir-fried vegetables were actually cooked. The most likely recipe can be estimated from the recorded audio-visual data using our system, which was constructed in the same setting as in Section 4.6.2. The left panel of Figure 4.6 shows the results of event classification by the audio-visual classifier and the center panel shows procedure recognition from classified events using the HHMM-based recipe model. The accuracy of event classification was only 63%, lower than that of the experiment described in Section 4.6.1. Accuracy was reduced by inclusion in this real scene of actions such as placing ingredients on the cutting board. Outputs of the event classifier include small segments that conflict with this recipe for stir-fried vegetables. These segments, however, were removed by procedure recognition. These results suggest that information about the recipe complements audio-visual data to recognize cooking procedures.

4.7 Prototype of a cooking support robot

We also built a prototype of a cooking support system. Our target users were beginning cooks who are generally familiar with recipes that include ingredients, but do not know details of recipes, such as how long ingredients should be heated and the size they should be before heating. Rather than learning the details of a recipe, these beginners may improvise, which may result in failure and interferes with learning the correct way

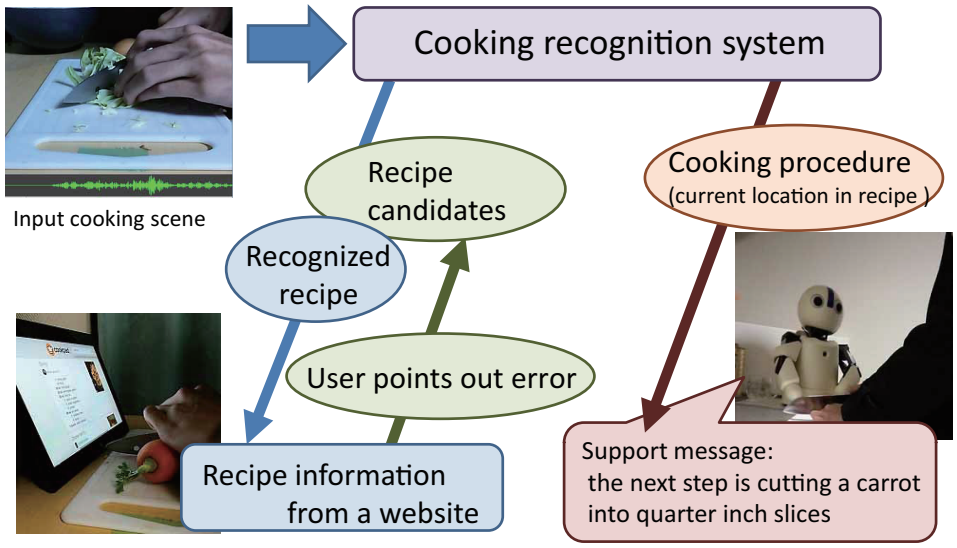


Figure 4.7: Cooking support system

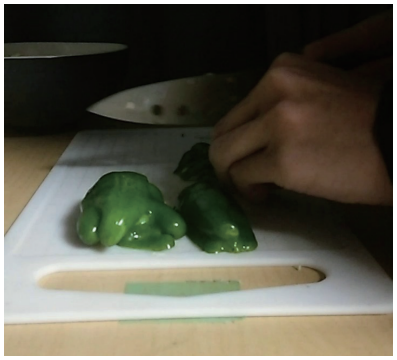
to cook. We therefore designed our system to resolve these problems and support user cooking.

Our system was designed to have three functions, as shown in Figure 4.7. The first was to generate candidate recipes from input recipes using audio-visual data and to display the web page corresponding to the most-likely candidate. The second was interactive recipe selection. When a user mistakenly selects a recipe, the system displays another candidate recipe, allowing the user to select the most appropriate recipe. These two functions are related to cooking recipe recognition. When the user cooks a dish, this system automatically displays the recipe site corresponding to the most-likely or selected recipe. This recipe, however, may be selected erroneously, since the accuracy of the recipe recognizer for the top-1 selected recipe is quite low, as described in Section 4.6.2. We therefore constructed such an interface using n candidates, which provided higher category accuracy and noise robustness. The third function is related to recognition of cooking procedures. This system recognizes procedures, as in Section 4.6.3, to extract support messages like "the next step is cutting carrots into quarter inch slices". This message is extracted from the top-1 recipe or a recipe selected by the user from candidate recipes. These three functions enable dishes to be cooked without explicit searching and to be cooked correctly based on recipes.

4.8 Conclusion of this chapter

This chapter presented a scene analysis system as an example of a PIE framework and focused on a multi-sensory approach and acquisition of knowledge. As a result, we showed that the PIE framework was applicable to an audio-visual multimodal cooking recognition system with CNN-based audio-visual integration and a recipe model represented by an HHMM automatically constructed from a website. Our first experiment showed that our audio-visual multimodal approach outperformed audio-only and visual-only approaches. Audio information especially supports visual information when the target action occurs out of site of the camera. Our second experiment showed that the top-n candidates of recipes yielded noise-robustness, which was important for real-world applications. The third experiment, extraction of an optimal cooking procedure necessary for a cooking support system, showed that our recipe model constructed from recipes could correct events inaccurately recognized from audio-visual data. Finally we showed an example in which our framework was applied to a cooking support system. This system provided interactive cooking support to beginning cooks.

Future work includes scaling up this system and its application to a more realistic cooking support robot. This chapter assumed that sound and video were recorded at the same time and that environmental noise was low. For example, sound in cooking videos on websites such as Youtube (`youtube.com`) are often removed or else these videos include considerable environmental noise such as background music and speaking. These factors must be considered prior to utilizing large-sized datasets at this website.

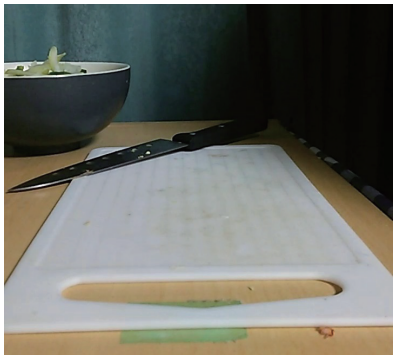


(1) Cutting green pepper

(2) Cutting onion

(3) Cutting carrot

(4) Cutting radish



(5) Baking

(6) No action

Figure 4.8: Recording of cooking events

Chapter 5

Bird Song Analysis

This chapter focuses on event layer in the PIE framework and tackles Issue 1. "How to extract attributes of an event", using an approach that differs from the approach used in the previous chapter. This chapter deals with a problem of co-dependency between separated sounds (also described in Chapter 2). To resolve this problem, this chapter proposes a new Spatial-Cue-Based Probabilistic Model (SCBPM) to identify and label a sound event considering spatial information. As an example of its applications, this chapter addresses bird song analysis, because groups of birds have their own territory and bird songs are strongly associated with spatial information on birds. This chapter considers the application of a semi-automatic annotation system for bird songs, and also addresses the problem of partial annotation, which is an application-dependent problem.

Applications to identify bird songs are required of bird researchers. In order to set more realistic problem, preliminary experiments were carried out by actual observation of wild bird with experts.

5.1 Introduction

Bird song analysis seeks to extract information from recorded bird songs. Bird songs are important in advertising territory and in courtship [53]. Localizing and identifying bird songs, especially for wild birds in the field, can assist in research on bird communication in ethology. Experiments studying bird song can be divided into two main types: laboratory experiments and field observations. Because their conditions can be well controlled, laboratory experiments have contributed to understanding of bird songs. However, field observations of bird songs are also essential, because birds may behave different in the

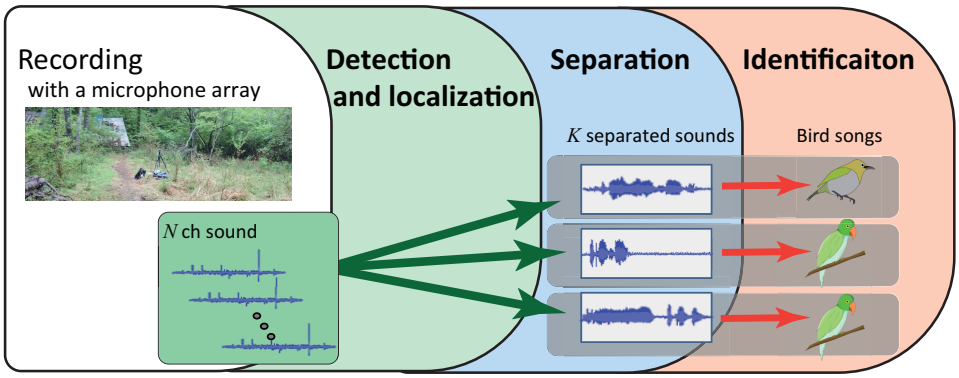


Figure 5.1: Cascade model for bird song identification

wild than in the laboratory. Analysis of wild bird songs requires annotation tasks, with the conventional method of annotation being manual and scientists observing birds and listening to bird songs in the field. This process is time consuming and laborious and frequently yields consistent results. Several recent studies, using single-channel and binaural microphones, have evaluated bird song identification tasks [54, 55]. Arrays of three or more microphones have also been used to record bird songs in the field [56]. These microphone arrays provide a powerful system for sound source localization and separation. This chapter shows that microphone arrays are also useful also in bird song identification. Our ultimate goals include automatic bird song annotation and the development of a tool for sounds recorded with a microphone array.

This chapter proposes a new SCBPM, integrating sound source detection, localization and separation for sound source identification [57]. We applied this SCBPM to bird song identification and constructed a prototype of a semi-automatic annotation system to evaluate our model. Development of our model and system must consider three factors:

- Spatial information
- Modeling of the relationships among separated sounds
- Partially annotated data

Spatial information is an important cue in real-world applications, as spatially close sound sources are probably in the same category. For example, songs vocalized within a territory have a high probability of being from the same individual or species. Spatial information results from sound source localization. For example, the MUltiple SIgnal Classification (MUSIC) method [8] has been used to estimate the direction of arrival in

such situations. Because these results often include errors, we constructed a probabilistic generative model that included not only acoustic features but locations of sound sources using a von Mises distribution, which distributes sources as directions on a circle. By considering the locations of sound sources, this model enables sound source identification. Among the many other approaches to sound source localization are distributed microphones [58] and 3D localization [59]. As the first step, we adopted 2D localization using a single microphone array and modeling by the von Mises distribution. Although this chapter focuses on spatial information, sound source identification has another aspect: temporal information. Among the approaches available to address temporal information are non-negative matrix factorization (NMF)-based methods [60, 61, 62], weighted finite state transducer (WFST)-based methods [63], spectrogram-based methods [64, 65], hidden Markov model (HMM)-based methods [66], and a method using a Pitman-Yor Process to capture temporal changes [67]. We are not concerned here with temporal changes, like long phrases of bird songs. The combination of these methods and our approach would be promising.

The second factor to be considered is modeling for spatial relationships among separated sounds. As mentioned above, the spatial information can be represented using a von Mises distribution. In considering spatial information, the spatial relationships among two or more sound sources are important. One model for such relational data is a stochastic block model (SBM), a probabilistic model for relational data in social networks that can be applied to network community analysis and relational data clustering [68]. Relational data clustering is data clustering that utilizes relationships among two or more objects. Although we have targeted these relationships, the SBM cannot directly apply our tasks because it deals with discrete distributions, where as acoustic signals and directions are continuous data. Therefore, we derived a new specialized model to represent spatial relationships among sound sources.

The third factor is a partially annotated dataset. Our target is a semi-automatic system, because completely automatic annotation is practically impossible, as it requires that the accuracy of an identifier be 100%. Therefore, we sought to construct a human-in-the-loop system, in which specialists annotate some of the data and the remainder is automatically annotated using a sound source identifier. Subsequently, specialists can adjust the estimated labels and retrain the sound source identifier. This task can be formalized as semi-supervised learning using mixtures of annotated and non-annotated data. Our generative modeling approach can deal with non-annotated data

by regarding these data as missing; for example, parameter training using an expectation maximization (EM) algorithm [69]. We constructed a prototype annotation system using semi-supervised training to effectively utilize non-annotated data.

This chapter describes, the construction of the SCBPM, considering these three factors and a prototype system to evaluate our model. We also performed experiments using a real bird song dataset, showing that our method outperformed a conventional cascade model. Section 5.2 introduces the conventional cascade model; Section 5.3 describes the SCBPM; and Section 5.4 explains our system using the SCBPM. Section 5.5 shows the evaluation of our model using real data and Section 5.6 concludes this chapter.

5.2 Conventional cascade model

Figure 5.1 shows a conventional cascade model, previously described in Section 2.2, in which recorded sound is processed in order of detection, localization, separation, and identification. These processes are performed using separate methods. For example, the MUSIC method is used for detection and localization, the beam-forming method for separation, and an identifier using GMM (Gaussian mixture model), based on the acoustic features of separated sounds, for identification.

Detection and localization require estimates of the number of acoustic events and of the duration and direction of each event. These can be accomplished using the MUSIC method; its extended method works well under noisy situations, such as outdoors [12]. The MUSIC method computes the power of the signal sub space for each direction, called the MUSIC power spectrum. Because the signal and noise subspaces are orthogonal, this method achieves noise robustness. The number of candidate acoustic events, which equals the number of dimensions of signal subspace, is a parameter of the MUSIC method. Direction of arrival can be estimated by detecting peaks of the MUSIC power spectrum. The number of acoustic events can also be determined by thresholding peaks of the power. Another threshold of the difference between the direction related to the current frame and the direction related to the previous frame is used to track acoustic events temporally. These thresholds are also parameters of the MUSIC method and should be manually selected.

Sound source separation extracts the target sound source from a mixture of sound sources. The beam-forming method, using a transfer function between a sound source and a microphone array for each direction, is one type of sound source separation. The

geometric highorder decorrelation-based source separation (GHDSS) method [9] is an extension of beam-forming; it considers higher-order decorrelation of separated sounds, thereby effectively suppressing directional noise sources. Because our target is wild bird songs, many directional noise sources exist outdoors; we therefore utilized the GHDSS method for sound source separation.

Sound source identification in the cascade model is used to estimate the class, such as bird species, from the separated sounds. This can be formalized as computing the a posteriori probability of a class c from an acoustic feature $\mathbf{x}$ extracted from the separated sound by assuming an acoustic model. GMM is a naive, but effective, acoustic model, applicable to many extensions. In the cascade model, separated sounds are identified independently using a GMM. To use a GMM, we have to determine an acoustic feature. We utilized a frequency spectrogram computed by short-term Fourier transform (STFT) of separated sounds and constructed a 32-dimensional acoustic feature for each time t by dimensional reduction using principal component analysis (PCA). This dimensional reduction resulted in faster computations for parameter learning and identification on a GMM. The method of computing this acoustic feature is detailed in Section 5.5. We assumed that an acoustic feature $\mathbf{x}$ at time t of a sound source belonging to class c was generated from GMM of class c:

$$\begin{aligned} p(\mathbf{x}, s_c, c) &= p(\mathbf{x} \mid s_c) P(s_c \mid c) P(c) \\ &= \mathcal{N}_{s_c}(\mathbf{x}) P(s_c \mid c) P(c) \end{aligned} \tag{5.1}$$

where s_c is a subclass of class c, and $\mathcal{N}(\cdot)$ is a multi-variate Gaussian distribution. This model assumes the data were i.i.d., in that all acoustic features were generated independently by Eq.(5.1).

By dealing with the probabilistic variable c, we could compute probabilities on this model, whether or not c is fixed. This property achieved semi-supervised training with a maximum a posteriori (MAP)-EM algorithm. This model also yields a MAP-estimated formula for sound source identification.

$$c^* = \operatorname*{argmax}_c P(c \mid \mathbf{x}) \tag{5.2}$$

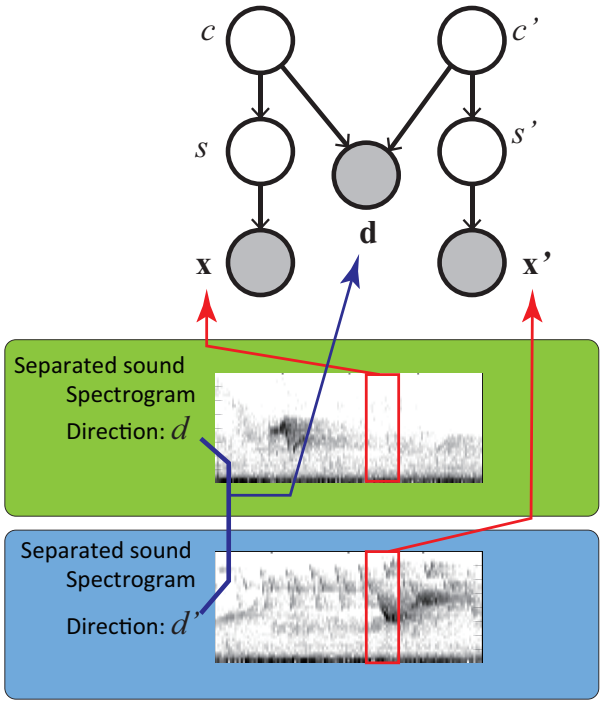


Figure 5.2: SCBPM for two sound sources, showing that simultaneous acoustic features ($\mathbf{x}$, $\mathbf{x}'$) depend on each other via their direction $\mathbf{d}$.

5.3 Proposed model: SCBPM

In this section, we describe a model for sound source identification that considers spatial information about sound sources.

5.3.1 Proposed model: SCBPM

Conventional GMM-based sound source identification is described in Section 5.2. The probabilistic variable related to a class label c is independent of each sound source k_t at time t. We introduced a dependency of sound sources based on their relative location. Figure 5.2 is an example, in which there are two sources at time t.

The direction d_{t,k_t} of a sound source k_t at time t can be computed by sound source localization methods such as the MUSIC method ($0 \leq d_{k_t} < 2\pi$). Considering multiple sound sources, K_t is the number of simultaneous events at time t and can be computed by the MUSIC method with a threshold; hence K_t is fixed in this model. A vector of directions of events at time t can be defined as $\mathbf{d}_t \stackrel{\text{def}}{=} (d_{t,1}, d_{t,2}, ..., d_{t,k_t}, ..., d_{t,K_t})$ $(1 \leq$

$k_t \leq K_t$).

Acoustic features $\mathbf{x}_{k_t}$ of sound sources k_t at time t can also be computed by sound source separation methods, such as GHDSS in Section 5.2. Because we were not concerned with temporal dependencies, the subscript t has been omitted from the following discussion. Using both the directions and the acoustic features, the SCBPM can be expressed as:

$$\begin{aligned} p(\mathbf{X}, \mathbf{d}, \mathbf{s}, \mathbf{c}) =& p(\mathbf{d} \mid \mathbf{c}) \prod_{k=1}^{K} p(\mathbf{x}_k \mid s_k) P(s_k \mid c_k) P(c_k) \\ =& p(\mathbf{d} \mid \mathbf{c}) \prod_{k=1}^{K} p(\mathbf{x}_k \mid s_k) P(s_{c_k} \mid c_k) P(c_k) \\ =& p(\mathbf{d} \mid \mathbf{c}) \prod_{k_t=1}^{K} \mathcal{N}_{s_{c_k}}(\mathbf{x}_k) P(s_{c_k} \mid c_k) P(c_k) \end{aligned} \tag{5.3}$$

$$\begin{aligned} p(\mathbf{d} \mid \mathbf{c}) = & \prod_{c_i = c_j, i \neq j} p(d_i, d_j \mid c_i = c_j) \\ & \prod_{c_i \neq c_j, i \neq j} p(d_i, d_j \mid c_i \neq c_j) \end{aligned} \tag{5.4}$$

$$p(d_i, d_j \mid c_i = c_j) = f(d_i - d_j; \kappa_1) \tag{5.5}$$

$$p(d_i, d_j \mid c_i \neq c_j) = f(d_i - d_j + \pi; \kappa_2) \tag{5.6}$$

$$f(d; \kappa) = \frac{\exp(\kappa \cos(d))}{2\pi I_0(\kappa)} \tag{5.7}$$

where $f(d; \kappa)$ refers to the von Mises distribution, $I_0(\kappa)$ is the modified Bessel function of order 0 and κ is a measure of concentration. When κ equals zero, $f(d; \kappa)$ is uniform. When the two sound sources are close and belong to the same class, $p(d_i, d_j \mid c_i = c_j)$ has a high probability, as shown by Eq.(5.5). In contrast, when the two sound sources are far apart and belong to different classes, $p(d_i, d_j \mid c_i \neq c_j)$ has a high probability, as shown by Eq.(5.6) [1] . To consider more than two sound sources, $p(\mathbf{d} \mid \mathbf{c})$ is defined as

[1]The property, $p(d_i, d_j \mid c_i \neq c_j)$ can be expressed as a von Mises distribution, in which the first argument is $d_i - d_j + \mu$ such that $\mu = \pi$. Although we utilized $\mu = \pi$, alternative distributions can be considered. For example, setting the number of sound sources at three results in $\mu = \pm\pi/3$, yielding a multi-modal distribution with additional parameters required to control this distribution. Alternatively, μ may be trained from the data, but this method requires a larger dataset to determine μ.

the product of a combination of sound sources, as shown by Eq.(5.4). This model has two parameters, κ_1 and κ_2, in addition to the GMM parameters described in Section 5.2. These parameters are also trained from data (described in Section 5.3.2).

Because c_i and c_j depend on each other, the i.i.d. assumption of the conventional GMM identifier is violated. This changes the MAP-estimation formulation from Eq.(5.2) to:

$$\begin{aligned}\mathbf{c}^* &= \underset{\mathbf{c}}{\operatorname{argmax}}\, P(\mathbf{c} \mid \mathbf{x}, \mathbf{d}) \\ &= \underset{\mathbf{c}}{\operatorname{argmax}}\, p(\mathbf{x} \mid \mathbf{c}) P(\mathbf{d} \mid \mathbf{c}) P(\mathbf{c})\end{aligned} \tag{5.8}$$

Unlike conventional methods, classes $\mathbf{c}^*$ of sound sources k_t at time t are estimated simultaneously. The point of this formulation is only the addition of a correction factor. Therefore, implementation is easy and, if a conventional model has been trained, only this factor must be computed.

5.3.2 Parameter training for the SCBPM

In this section, we introduce training that considers spatial information.

An EM algorithm requires the computation of the expected probability of a subclass in a dataset. This expectation N_s can be computed from a posteriori probabilities of subclass s as:

$$N_s = \sum_t \sum_{k_t} \gamma_{s,t,k_t}$$

$$\gamma_{s,t,k_t} = P(s_{t,k_t} = s|\mathbf{X}, \mathbf{D})$$

where s_{t,k_t} is a probabilistic variable of a subclass related to an acoustic feature of a sound source k_t at time t, $\mathbf{X}$ is a set of all acoustic features and $\mathbf{D}$ is a set of directions of all sound sources. $P(s_{t,k_t} = s|\mathbf{X}, \mathbf{D})$ is computed over the SCBPM. Note that $P(s_{t,k_t} = s|\mathbf{X}, \mathbf{D})$ depends not only on an acoustic feature of a sound source k_t at time t but on other sound sources at time t according to Figure 5.2 and the properties of a Bayesian network. When, for simplicity, we set an acoustic feature of a sound source k_t at time t at $\mathbf{x}$ and there is another acoustic feature x' at time t, then $P(s_{t,k_t} = s|\mathbf{X}, \mathbf{D})$ can be

described as:

$$P(s_{t,k_t} = s|\mathbf{X}, \mathbf{D}) \propto \sum_{c,c'} P(s|c,x)p(d,d'|c,c')P(c)P(c')p(x'|c') \tag{5.9}$$

and $p(x'|c') = \sum_{s'} p(x'|s')P(s'|c')P(c')$. Conventional GMM can be used to compute $P(s|c,x,d)$ with other defined factors.

Parameters of von Mises distributions in our model, κ_1 (Eq.(5.5)) and κ_2 (Eq.(5.6)) can also be trained by an EM algorithm. These equations can be derived from an EM algorithm for a mixture of von Mises distributions [70]. Updated κ_1 can be computed as:

$$U_{c=c'} = \sum_t \sum_{x,x'} \sum_{c=c'} cos(d-d')P(c \mid d, x_t)P(c' \mid d', x'_t) \tag{5.10}$$

$$V_{c=c'} = \sum_t \sum_{x,x'} \sum_{c=c'} P(c \mid d, x_t)P(c' \mid d', x'_t) \tag{5.11}$$

$$\kappa_1^{(new)} = A^{-1}\left(\frac{U_{c=c'}}{V_{c=c'}}\right) \tag{5.12}$$

$$\tag{5.13}$$

where $\kappa_1^{(new)}$ is updated κ_1. $U_{c=c'}$ and $V_{c=c'}$ can be computed over the model using the sum of all possible combinations of acoustic events ($\mathbf{x}$ and $\mathbf{x}'$) at the same time, such that $c = c'$. $A(x)$ can be defined as:

$$A(x) = \frac{I_1(x)}{I_0(x)}$$

where $I_0(x)$ and $I_1(x)$ are modified Bessel function of orders zero and one, respectively. The inverse function, $A^{-1}(x)$ can be approximated by $A^{-1}(x) \approx \frac{x(2-x^2)}{1-x^2}$ [71]. The update for κ_2 is the same except for substitution of $c \neq c'$ for $c = c'$.

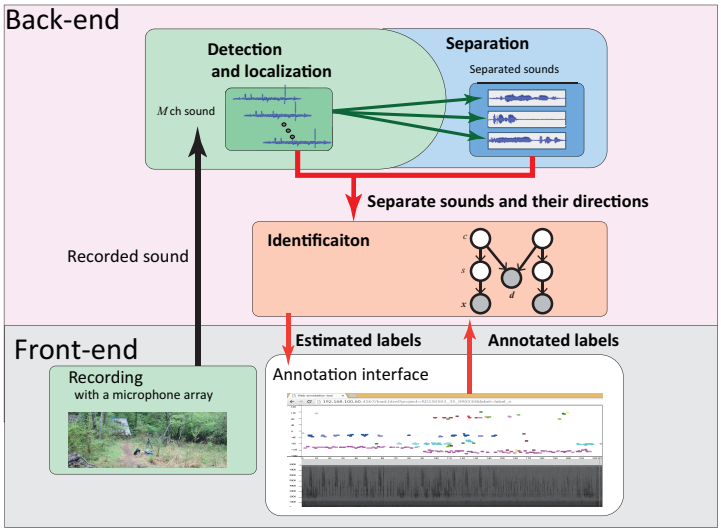


Figure 5.3: Prototype system (a large scale interface snapshot is shown in Figure 5.7)

5.4 Prototype of a semi-automatic annotation system

To evaluate our model, we constructed a prototype system, consisting of back- and front-end parts (Figure 5.3). The back-end part includes sound source detection, localization, separation and identification, which are performed automatically in principle. Sound source detection, localization and separation are implemented using HARK[2] [72], which includes MUSIC and GHDSS modules. These modules require a transfer function between a sound source and microphones, which can be computed by recording a time-stretched pulse[3]. Sound source identification is implemented as described in Section 5.3.

The front-end part, which consists of recording and annotating devices, must be performed manually. For recording, we used a microphone array called Microcone [4], comprised of seven microphones. This array was mounted on a tripod to capture bird songs, mainly from birds on trees, and connected to a laptop computer for recording (16-bit precision, 16kHz sampling). Figure 5.4 shows this recording system. The annotation module presents a panel showing the results of bird song analysis by the back-end part.

[2]Honda Research Institute Japan Audition for Robots with Kyoto University
[3]http://www.hark.jp/
[4]http://www.dev-audio.com/

Figure 5.4: Recording system with a microphone array

The horizontal and vertical axes of the panel represent time and sound source direction, respectively. The colored lines in the panel indicate bird song events, with the colors representative of different bird species. The sound spectrogram, a visual representation using three axes, for time, frequency and amplitude, is also displayed for bird song annotation specialists.

When actual recorded sound is input into the system, the system estimates labels using a default model that may not be trained sufficiently. After correction of these estimated labels by specialists, the system can re-train the model using both the annotated and re-estimated labels. Iterations of this process yield a sufficiently-trained identifier with well-annotated data.

5.5 Experiments

To evaluate the SCBPM, we constructed two datasets of recordings at different places.

Dataset (A) consisted of bird songs recorded in an urban park in Aichi, Japan on the morning of May 5, 2013, a sunny day during the bird song season. Application of the MUSIC method described in Section 5.2 to this recorded sound yielded 54 automatically extracted events. The threshold value of the MUSIC power was adjusted so that each event represented as much of a phrase of bird song as possible. The separated sounds can also be computed, with these being good cues for annotation. To construct the dataset,

Table 5.1: Dataset (A): Bird song events, number of events, and colors in Figure 5.6

Label (species)	# of events	color
Narcissus flycatcher	5	red
Japanese white-eye	7	cyan
Brown-eared bulbul (A)	12	blue
Brown-eared bulbul (B)	13	yellow
Other	17	green

Table 5.2: Dataset (B): Bird song events, number of events, and colors in Figure 5.8

Label (species)	Code	# of events	color
Pacific-slope flycatcher	PSFL	7	green
Spotted towhee	SPTO	8	red
Nashville warbler	NAWA	12	blue
Black-headed grosbeak	BHGR	10	cyan
Orange-crowned warbler	OCWA	4	yellow
Cassin's vireo	CAVI	90	magenta
Unknown bird song	Unk.	6	dark green
Others	Oth.	3	dark red

all events were manually annotated. Figure 5.6 shows a snapshot of our prototype system related to this dataset. We used five types of labels (Table 5.1), which were annotated manually and are regarded as correct, as well as corresponding to the domain of c described in Sections 5.2 and 5.3.

Dataset (B) consisted of bird songs recorded for 4 min on the morning of May 9, 2013, in a mixed conifer-oak woodland forest in California in the United States. This set consisted of 140 events annotated as in Dataset (A). Figure 5.7 shows a snapshot related to this dataset. We used eight types of labels (Table 5.2). This dataset was sparser and with less overlap than Dataset (A).

Our method was evaluated by 10-fold cross-validation for each dataset. Each dataset was divided into ten periods of equal length [5] . The $10 \times r$ periods were labeled data and the others were unlabeled data. An event at a border was included in the period that encompassed more than half of the event. Accuracy was computed by comparing estimates of unlabeled data with manual annotations. This test was performed for each r from 0.1 to 0.9 at intervals of 0.1. Semi-supervised training included the weights of

[5]Some events were rare in our datasets. These events often yielded highly varying results. These experiments did not include special preprocessing as it did not pose a significant problem.

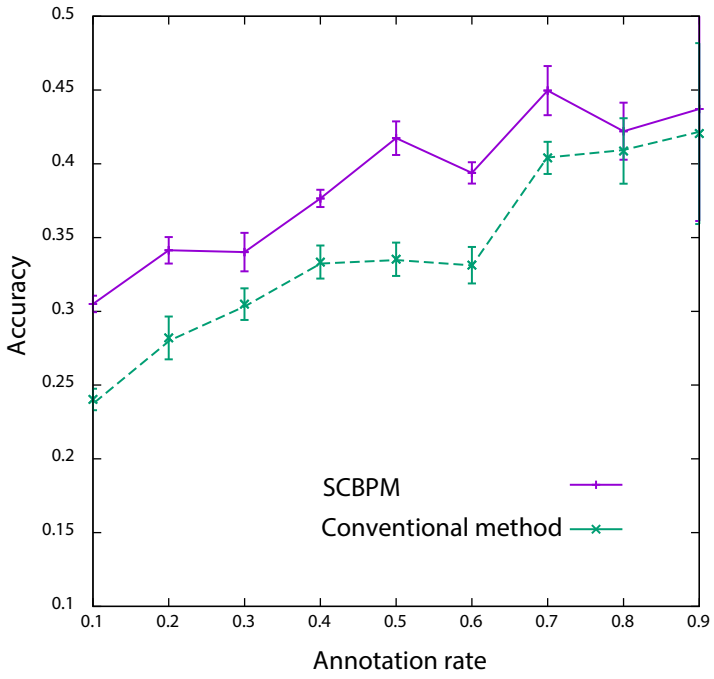


Figure 5.5: Comparative accuracy of our model and a conventional cascade model for Dataset (A).The error bars represent $\pm\sigma$ (standard deviations by 10-fold cross-validation).

labeled and unlabeled data [69]. The weight of labeled data was set at 1.0 and the weight of unlabeled data was set at 0.1 for Dataset (A) and 0.001 for Dataset (B).

Acoustic features were computed from separated sounds encoded with 16kHz sampling. Vectors of 41 dimensions were computed for each frame using STFT with 80 sample windows of 5 msec each and 40 overlaps of 2.5 msec each. These settings are often used by bird song annotators. Blocks were extracted with 100 frame windows and 90 frame overlaps, with each block represented as a 4100-dimension vector. The 32-dimension acoustic features $\mathbf{x}$ described in Sections 5.2 and 5.3 were computed from each block by PCA. The number of Gaussians in the GMM was determined by the Bayesian Information Criterion (BIC) [73] and set at 30 .

Figures 5.5 and 5.8 show the results related to Datasets (A) and (B), respectively. The SCBPM outperformed the conventional method for almost all r, showing that spatial information is a good cue for sound source identification.

The accuracy of this method was higher for Dataset (B) than for Dataset (A) (Figure 5.6, Figure 5.7). This may have been caused by the greater sound source separation in

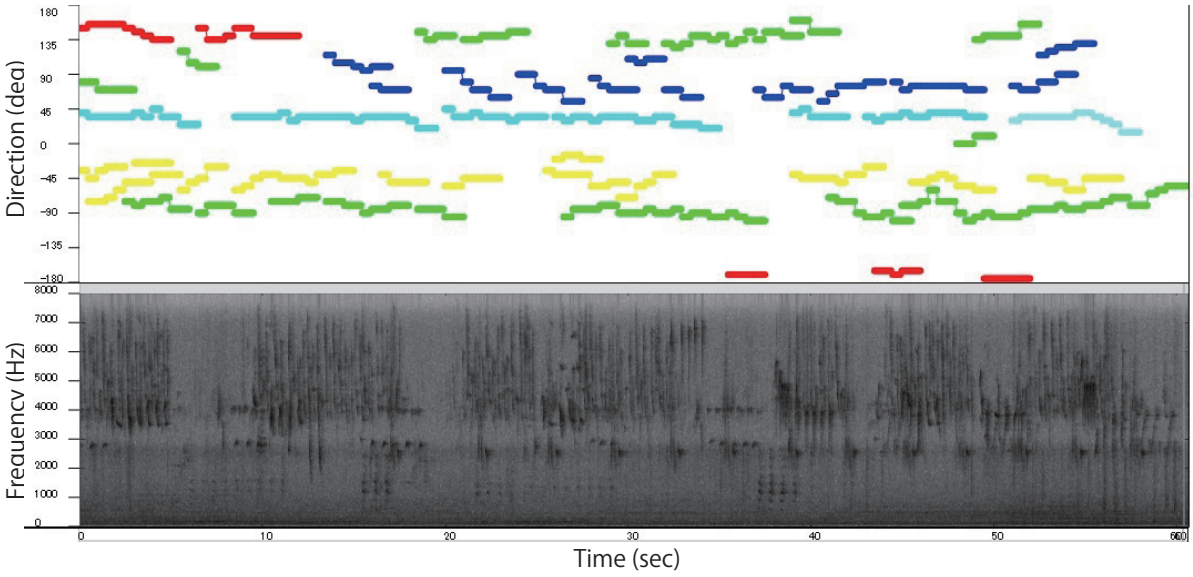


Figure 5.6: Dataset (A): One minute dataset recorded in Japan. The upper panel shows annotation by direction (y-axis) and time (x-axis), and the lower panel shows a spectrogram. Pairs of colors and bird songs are shown in Table 5.1. Brown-eared bulbuls (A) and (B) were different individuals, with different song features from each other; hence, they were assigned separate labels.

Dataset (B). Fewer simultaneous events occurred in Dataset (B) than in Dataset (A), resulting in good sound source separation and more accurate identification in the former.

The SCBPM outperformed the conventional method, especially when the annotated data ratio was > 0.6 (Figure 5.7). This finding suggests that the SCBPM is empowered by spatial information, but that it is difficult to identify sound sources based on spatial information alone. As spatial information compensates for the lack of sound information, the performance of a base classifier should be high enough (in this case, accuracy was over 0.6, with $r \geq 0.5$) to make use of spatial information effectively.

The ability of the SCBPM to consistently outperform the conventional method indicates that the SCBPM was effective when acoustic events were overlapping and dense. In the absence of overlapping events, the performances of the SCBPM and the conventional GMM were identical. Density was related to the basic assumption of the SCBPM, that nearby sound sources belong to the same category. This finding also suggested that spatial relationships are more important when there are many sound sources. Tables 5.3 and 5.4 show the confusion matrices of the conventional method and SCBPM for Dataset(A) and $r = 0.2$. These results show that the SCBPM can correctly estimate

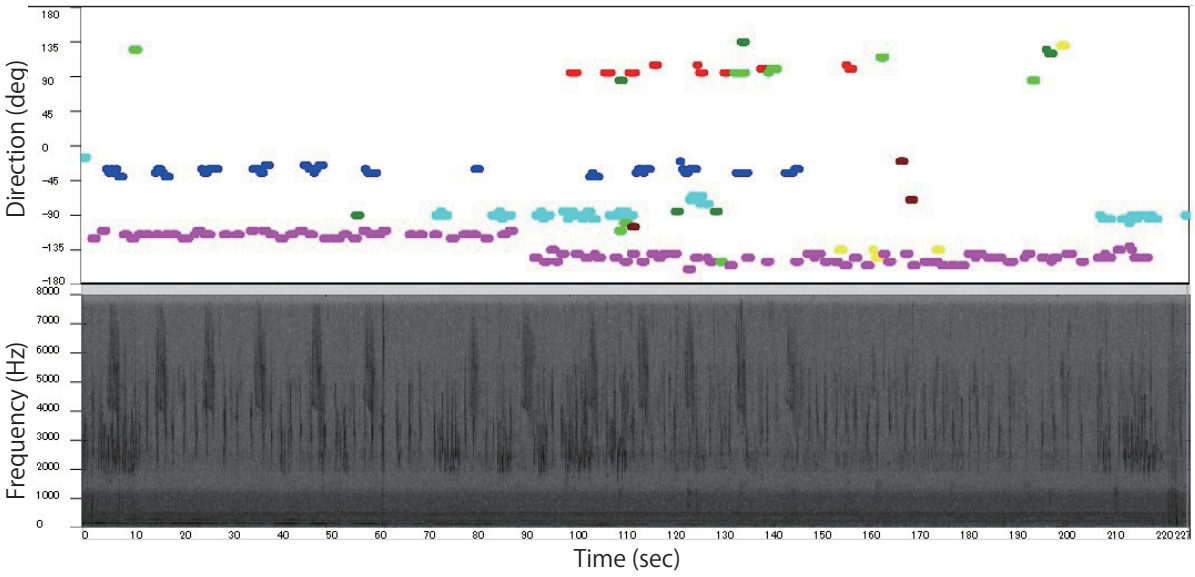


Figure 5.7: Dataset (B): Four minute dataset recorded in the USA. The upper panel shows annotation by direction (y-axis) and time (x-axis), and the lower panel shows a spectrogram. Pairs of colors and bird songs are shown in Table 5.2.

two additional events, a BHGR and a NAWA event.

Another topic of interest is scalability and more detailed evaluation related to other bird species. Both datasets in this study were small. Although other bird song datasets have been compiled [54, 55], these sounds were recorded with single or binaural microphones. This study showed the effectiveness of spatial information utilizing a microphone array. Future work includes the development of a larger bird song dataset recorded with microphone arrays and further research related to scalability.

5.6 Conclusion of this chapter

This chapter presented bird song analysis based on semi-automatic annotation. We proposed a new model, SCBPM, for integration of sound source detection, localization, separation and identification by expanding robot audition technologies. The SCBPM integrated these functions based on a generative model that included both acoustic features and the location of sound sources. We utilized the dependency derived from spatial relationships among sound sources to train and estimate this model by MAP estimation and an EM approach. Because data annotation of acoustic signals in the wild is

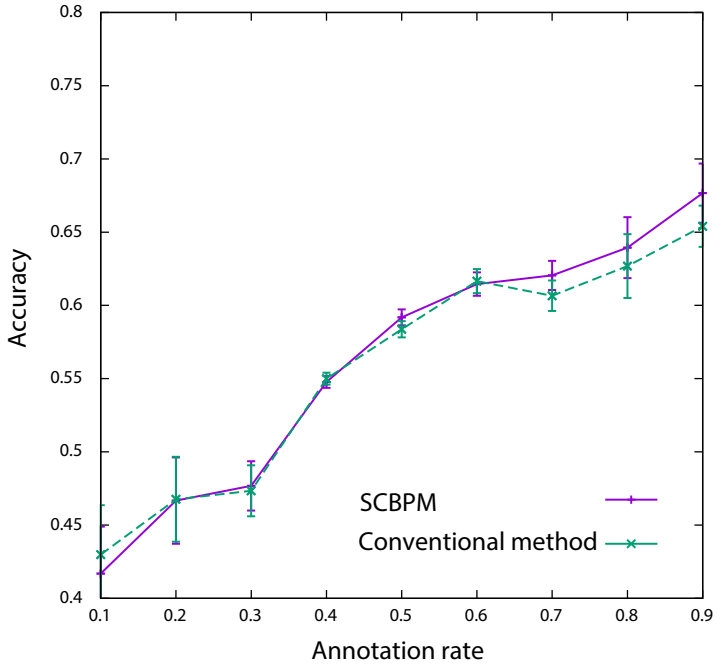


Figure 5.8: Comparative accuracy of our model and a conventional cascade model for Dataset (B). The error bars represent $\pm\sigma$ (standard deviations by 10-fold cross-validation).

problematic, we also proposed a semi-automatic annotation approach to address this problem. We constructed a prototype system for semi-automatic bird song annotation based on the SCBPM approach, and showed that the system was more accurate in identification than a conventional method based on robot audition (i.e. a cascade system). Future work application includes scaling up this system and applying it to more realistic annotation by specialists to obtain feedback from them.

Table 5.3: Confusion matrix of the conventional method (Dataset(B), $r = 0.8$). The columns and rows indicate collected and estimated labels corresponding to the codes in Table 5.2, respectively.

	PS FL	SP TO	BH GR	NA WA	OC WA	CA VI	Unk.	Oth.
PSFL	**4**	0	1	3	2	0	0	0
SPTO	0	**0**	0	0	0	0	0	0
BHGR	0	0	**1**	0	0	0	0	0
NAWA	0	0	0	**0**	0	0	0	0
OCWA	0	0	0	0	**0**	0	0	0
CAVI	0	0	0	0	0	**0**	0	0
Unk.	0	0	0	0	0	0	**0**	0
Oth.	0	0	0	0	0	0	0	**0**

Table 5.4: Confusion matrix of the SCBPM (Dataset(B), $r = 0.8$)

	PS FL	SP TO	BH GR	NA WA	OC WA	CA VI	Unk.	Oth.
PSFL	**4**	0	0	2	1	0	0	0
SPTO	0	**0**	0	0	1	0	0	0
BHGR	0	0	**2**	0	0	0	0	0
NAWA	0	0	0	**1**	0	0	0	0
OCWA	0	0	0	0	**0**	0	0	0
CAVI	0	0	0	0	0	**0**	0	0
Unk.	0	0	0	0	0	0	**0**	0
Oth.	0	0	0	0	0	0	0	**0**

Chapter 6

Web Session Log Analysis

This chapter focuses on the plan and intention layers of the PIE framework and addresses Issue 2, "How to consider relationships between events", using intention and plan recognition. Chapter 4 describes the use of HHMM for this task, whereas this chapter adopts probabilistic context-free grammars (PCFGs) to allow more flexible plans. This chapter proposes a method to deal with *incomplete data situation* and evaluates this method by web session log analysis.

6.1 Introduction

A main task of web session log analysis is determining visitor intention from web session log data. This task has been performed in various fields including computer security [74], web robot detection [75] and web usage mining [76]. In computer security, for example, the detection of attacks on a website is of a prime importance. The problem has been formulated as determination of whether visitors to a website do or do not intend to attack that website. Detection methods using HMMs have been proposed and proven effective at addressing this problem [77, 78]. Plan recognition has been utilized to detect intrusion on websites [79], a method we also utilized.

Plan and intention recognition in artificial intelligence have been regarded as an inverse of planning and applied to services that require the determination of users' plans and intentions from their actions, such as human-computer collaborations [80], intelligent interfaces [81] and intelligent help systems [82]. This chapter addresses plan and intention recognition using PCFGs, which are widely used as models not only in natural language processing but in analyzing symbolic sequences in general. In a simple PCFG

model for plan and intention recognition, a symbolized action sequence constructed by a user is regarded as a sentence generated by a mixture of PCFGs, with each component PCFG describing a plan for a specific intention. We call this model *a mixture of PCFGs*. In this setting, intention recognition must infer intention from a sentence as the most likely start symbol of some component PCFG in the mixture. In contrast, plan recognition computes the most likely parse tree representing a plan for the intention.

The problem with this simple model is that sentences generated by a grammar alone have a non-zero probability. Therefore, the probability of non-sentences is always zero, making it impossible to extract information contained in incomplete sentences by considering their probabilities and (incomplete) parse trees, despite their frequent occurrence in real data. To overcome this problem, we propose to generalize the probability and parse trees of sentences to those of incomplete sentences. Consider for example a *prefix* of a sentence, i.e., an initial substring of the sentence. This is one type of incomplete sentence, with data often observed at the start but before the completion of an observation, such as the medical records of a patient who is receiving treatment[1].

In plan recognition, our proposal enables the extraction of the most likely plan from incomplete data, providing important information about the user's intentions. When applied to a website, such as in determining a visitor's plan from his/her actions, such as clicking links, the discovered plan would reveal a website structure that matches the visitor's intentions. However, to our knowledge, no grammatical approach to date has extracted a plan, much less the most likely plan from incomplete sentences. This chapter shows the possibility of extracting the most likely plan from incomplete sentences.

Regarding intention recognition, it should be possible to directly determine intention from actions using feature-based methods, such as logistic regression and support vector machines. However, these methods are unable to utilize structural information represented by a plan behind the action sequence. We experimentally demonstrate the importance of structural information contained in a plan and compare our method with these feature-based methods in web session log analysis.

In our web session log analysis, action sequences recorded in session logs are basically regarded as complete sentences in a mixture of PCFGs. However, we consider three types of *incomplete data situations*. The first is an *online situation* designed to navigate

[1]The probability of a prefix in a PCFG is defined also the sum of probabilities of infinitely many sentences extending it. This prefix can be determined by solving a set of linear equations derived from the CFG [83]. In addition, prefix probability can be computed by method based on probabilistic Earley parsing [84].

web surfers to a target web page, by for example, displaying links appropriate to their intentions. The second is *unachieved visitors*, who quit a website for some reason prior to achieving their purposes. Because they do not fulfill their intentions, their action sequences should be considered incomplete sentences. The last type is the *cross-site situation*, in which a user visits several websites, a very likely situation during actual web surfing. In this situation an action sequence at one website is only part of the entire action sequence. Consequently an action sequence recorded at one website should be considered an incomplete sentence.

Notice that the first and second situations yield prefixes whereas the third results in the lack of both the starts and ends of sentences. This type of incomplete sentence is called an *infix*. This chapter primarily addresses the first and second situations, whereas the third situation and incomplete sentences like infixes are addressed in the Appendix. Our analysis is uniformly applicable to all situations made by visitors. Moreover, it can result in appropriate advertisements popping up in a timely manner during web surfing by detecting visitors' purposes and plans from action sequences recorded in web session logs.

In the following, we first review prefix probability computation. We next apply prefix probability computation via parse trees to web session log analysis and conduct an experiment on visitors' intentions and plans using real data sets.

To implement our approach, we used the logic-based modeling language PRISM [85, 86], which is a probabilistic extension of Prolog for probabilistic modeling. This chapter does not address its implementation and more general cases. For such information, please see the Appendix.

6.2 Prefix probability computation

In this section, we examine prefix computation for PCFGs. A PCFG $\mathsf{G}_{\mathbf{\Phi}}$ is a CFG G augmented with a parameter set $\mathbf{\Phi} = \bigcup_{N^i \in \mathbf{N}} \{\phi_r\}_{N^i}$ where $\mathbf{N}$ is a set of nonterminals, N^1 is a start symbol and $\{\phi_r\}_{N^i}$ is a set of parameters associated with rules $\{r \mid r = N^i \to \zeta\}$ for a nonterminal N^i where ζ is a sequence of nonterminal and terminal symbols. We assumed that the ϕ_r's satisfy $0 < \phi_r < 1$ and $\sum_{\zeta: N^i \to \zeta} \phi_{N^i \to \zeta} = 1$.

Algorithms already exist to compute prefix probabilities in PCFGs [83, 84]. We here briefly describe prefix probability. As previously stated, a *prefix* $\mathbf{v}$ is an initial substring of a sentence and the prefix probability $P_{\mathrm{pre}}^{N^1}(\mathbf{v})$ of $\mathbf{v}$ is an infinite sum of probabilities

of sentences extending $\mathbf{v}$:

$$P_{\text{pre}}^{N^1}(\mathbf{v}) = \sum_{\mathbf{w}} P_{\mathsf{G}}(\mathbf{vw})$$

where $\mathbf{w}$ ranges over strings such that $\mathbf{vw}$ is a sentence in G. In computing prefix probability, we use a PCFG $\mathsf{G_0} = \{\ \mathtt{s} \to \mathtt{s}\,\mathtt{s} : 0.4,\ \mathtt{s} \to \mathtt{a} : 0.3,\ \mathtt{s} \to \mathtt{b} : 0.3\ \}$, where "$\mathtt{s}$" is a start symbol and "$\mathtt{a}$" and "$\mathtt{b}$" are terminals. For example, by definition, the prefix probability $P_{\text{pre}}^{\mathtt{s}}(\mathtt{a})$ of prefix $\mathtt{a}$ can be written as follows:

$$P_{\text{pre}}^{\mathtt{s}}(\mathtt{a}) = P_{G_0}(a) + P_{G_0}(ab) + P_{G_0}(aa) + P_{G_0}(aab) + ...$$

This infinite series is guaranteed to converge in the setting of prefix probability computation [83, 84]. More practical and general calculation methods are described in the Appendix.

6.3 Action sequences as incomplete sentences in a PCFG

In these sections, we address the problem of identifying the purposes or intentions of visitors to a website based on their session logs. We first abstract a visitor's session log into a sequence of five basic actions: `up`, `down`, `sibling`, `reload` and `move`. The first two, `up` and `down`, indicate that the visitor moves with respect to a page in the parent directory or a subdirectory in the site's directory structure. The action "`sibling`" indicates that the visitor moves to a page in a subdirectory of the parent directory. The action "`reload`" indicates that the visitor requests the same page, and the action "`move`" indicates remaining miscellaneous actions. Moving between web pages is expressed by a sequence of basic actions. For example moving from /top/index.html to /top/child/a.html is a `down` action.

We consider an action sequence generated by a visitor who has achieved the intention as a complete sentence in a PCFG. We parsed it using CFG rules (Figure 6.1 (left)), which describe possible structures behind visitors' action sequences, and obtained a parse tree (Figure 6.1 (right)).

Because visitors to a website have different intentions, we captured their action sequences $\mathbf{w}$ in terms of a mixture of PCFGs $P(\mathbf{w} \mid N^1) = \sum_A P^A(\mathbf{w} \mid A)P(A \mid N^1)$,

S → Survey
Survey → Search Destination
Search → Down Up Search | Down
Destination → `sibling` Destination | `sibling`
Down → Down `down` | `down`
Up → Up `up` | `up`

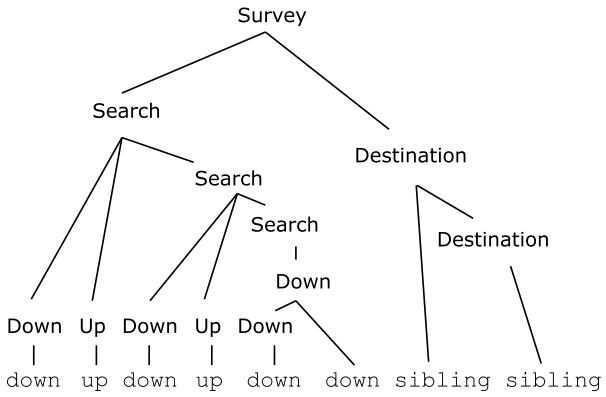


Figure 6.1: Example of CFG rules (left) and a parse tree using them (right)

where $P^A(\mathbf{w} \mid A)$ is the probability of $\mathbf{w}$ being generated by a visitor whose intention is represented by a nonterminal A and $P(A \mid N^1)$ is the probability of A being derived from the start symbol N^1. We call this A an *intention-nonterminal* and assume that there is unique rule $N^1 \to A$ for each intention-nonterminal A with a parameter $\theta_{N^1 \to A} = P(A \mid N^1)$.

Finally to make it possible to estimate visitor intentions from incomplete sequences, we replace a sentence probability $P^A(\mathbf{w} \mid A)$ in a mixture of PCFGs $P(\mathbf{w} \mid N^1) = \sum_A P^A(\mathbf{w} \mid A)P(A \mid N^1)$ by a prefix probability $P^A_{\text{pre}}(\mathbf{w} \mid A)$. We call this method the *prefix method*.

Suppose an action sequence of prefix $\mathbf{w}_k$ and length k. We estimate the most likely intention-nonterminal A^* for $\mathbf{w}_k$ by

$$A^* = \operatorname*{argmax}_A P^A_{\text{pre}}(\mathbf{w}_k)P(A \mid N^1) \tag{6.1}$$

where A ranges over possible intention-nonterminals. $P^A_{\text{pre}}(\mathbf{w}_k)$ is computed just like $P^{N^1}_{\text{pre}}(\mathbf{w})$ in the previous section.

6.4 Experiments

In this section, we empirically evaluated our prefix method and compared it with two existing methods: the PCFG method and logistic regression. The PCFG method applies a mixture of PCFGs to action sequences $\mathbf{w}_k$ by assuming that every sequence is a sentence. The most likely intention-nonterminal is estimated by substituting $P^A_{\text{pre}}(\mathbf{w}_k)$

Table 6.1: Result of clustering

Cluster (intention-nonterminal)	Features and major action
Survey	`up`/`down` moves in the hierarchy of a website
News	`up`/`down` moves in the hierarchy of a website + reload the same page
Survey(SpecificAreas)	access to the same layer
News(SpecificAreas)	access to the same layer + reload the same page
Other	others

in Eq.(6.1) with $P^A(\mathbf{w}_k)$.

We also compared the prefix method with logistic regression, which is a popular discriminative model that does not assume any structure behind data unlike the prefix and PCFG methods. For a fixed length k, the most likely visitor intention is estimated from $\mathbf{w}_k$ considered as a feature vector, in which the features are the five basic visitor actions introduced in Section 6.3.

6.4.1 Data sets and the universal session grammar

We prepared three data sets of action sequences by preprocessing the web server logs of University of the Saskatchewan (U of S), ClarkNet and NASA [87] in the Internet Traffic Archive [88]. We consider, solely for convenience, action sequences of length greater than 20 as sentences and exclude those with length greater than 30 as the computation of the latter is too costly. The three data sets from the U of S, ClarkNet and NASA contained 652, 4523 and 2014 action sequences respectively.

We next specified a CFG to build a mixture of PCFGs to apply to these data sets. This requires a determination of the numbers of intention-nonterminals. That is, we must decide how many intentions visitors have when visiting a website. We therefore clustered action sequences, by assuming that one cluster corresponds to one intention; i.e., the number of clusters is equal to the number of intention-nonterminals. We used a mixture of PCFGs again for clustering[2]. As a result, we obtained the five clusters which are listed in Table 6.1.

Finally we manually expanded the small CFG used for clustering into a large CFG called the *universal session grammar* that has five intention-nonterminals corresponding

[2] Clustering was performed using PRISM. We used a small CFG, containing 30 rules and 12 nonterminals, because clustering using a mixture of large PCFGs results in very high memory usage. To build this grammar, we merged similar symbols such as InternalSearch and Search in the universal session grammar shown in Table 6.2.

Table 6.2: Part of the *universal session grammar*

S → Survey
Survey → InitialSearch Destination EndSearch | Destination EndSearch
Destination→UpDownSearch Search | UpDownSearch
Search→InternalSearch Destination | InternalSearch

to five visitor clusters in Table 6.1. Some of the rules concerning Survey are listed in Table 6.2. The universal session grammar includes 102 rules and 32 nonterminals and reflects our observation that visitors have different action patterns during the initial, middle and final parts of a session.

6.4.2 Evaluation of the prefix method

We utilized the prefix method to estimate visitors' intentions from prefixes of action sequences and recorded the estimation accuracy while varying prefix lengths. As a reference method, we also applied a mixture of hidden Markov models (HMMs).

To prepare a teacher data set to measure accuracy, it is necessary to label each action sequence by the visitor's true intention, which is practically impossible. As a substitute, we defined a *correct top-intention* for an action sequence in a data set as being the most likely intention-nonterminal for the sequence, estimated by a mixture of PCFGs with the universal session grammar whose parameters are learned by the EM algorithm from the data set. This strategy seems to work as long as the universal session grammar is reasonably constructed.

In the experiment[3], accuracy was measured by five-fold cross-validation for each prefix length k ($2 \leq k \leq 20$). After parameter learning by a training data set, prefixes of length k are cut from the action sequences in the test set and their most likely intention-nonterminals were estimated and compared with their correct top-intention labels. Figure 6.2 shows the mean accuracy and standard deviation for each k.

Here Prefix denotes the prefix method, PCFG the PCFG method[4] and Log-Reg logistic regression analysis. For comparison, we also included HMM using a mixture of HMMs

[3]The experiment was performed on a PC with Core i7 Quad 2.67GHz, OpenSUSE 11.4 and 72GB main memory.

[4] A PCFG was applied to prefixes by considering them sentences. In this experiment, we found that the universal session grammar fails to parse at most two sequences for each data set, allowing us to ignore these sequences.

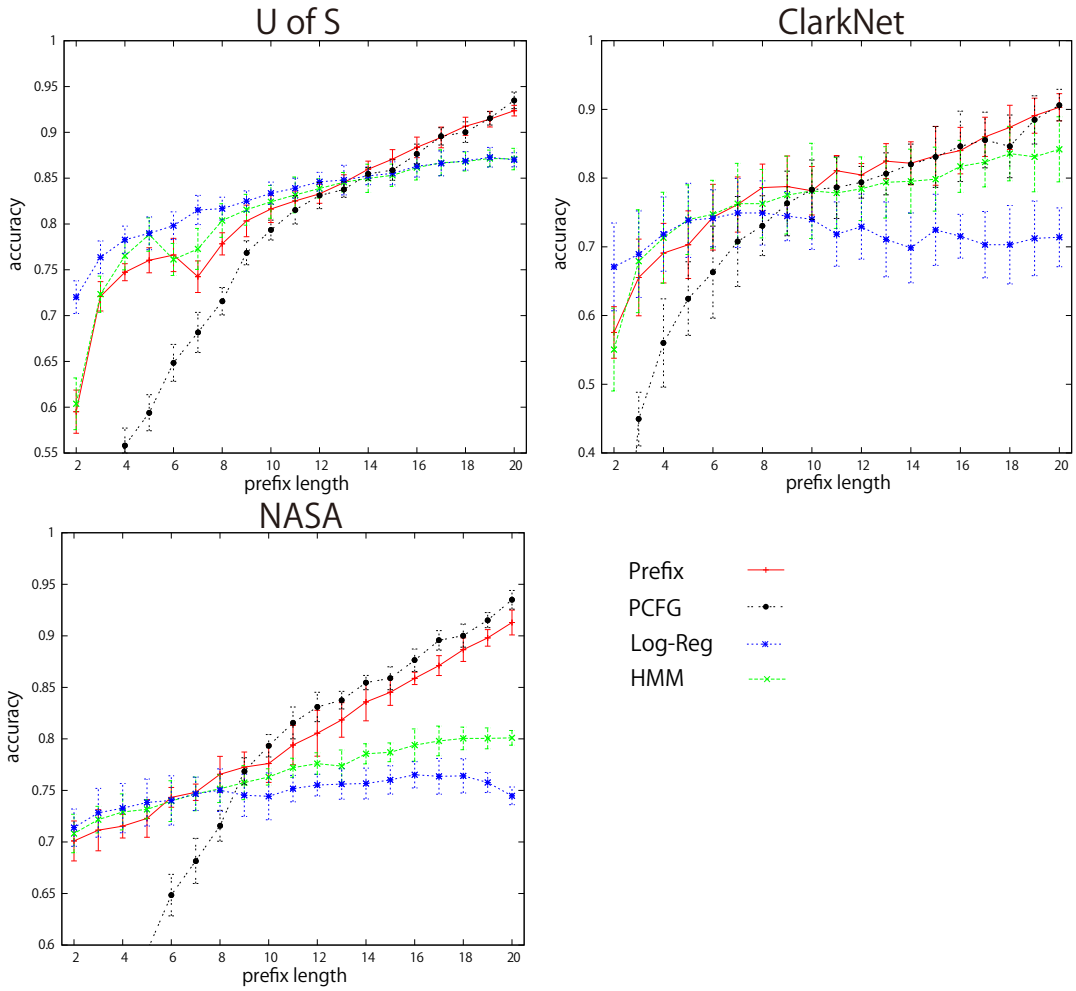


Figure 6.2: Accuracy for U of S, ClarkNet and NASA datasets

instead of a mixture of PCFGs[5] [6].

Figure 6.2 clearly demonstrates that the prefix and PCFG methods outperformed logistic regression and HMM when the prefix was long. Actually all differences for prefix length $k = 20$ in the graph were statistically significant ($p < 0.05$ by t-tests). We also found that, as prefixes shortened, the PCFG method rapidly deteriorated, although its performance was comparable to logistic regression and HMM.

We would like to emphasize that our approach can produce a most-likely plan for the

[5]We used a left-to-right HMM, with the number of states varying from 2 to 8. Figure 6.2 shows only the highest accuracy for each k. Because logistic regression only accepts fixed length data, we analyzed 19 logistic regression models, one for each length k ($2 \leq k \leq 20$).

[6]We used PRISM to implement mixtures of HMMs and of PCFGs and to compute prefix probability. To implement logistic regression, we used the 'nnet' package of R.

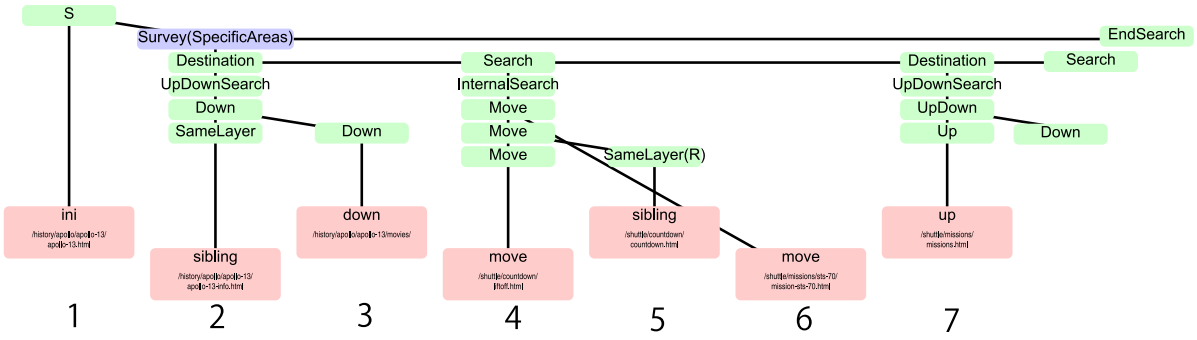


Figure 6.3: A prefix parse tree for an action sequence in the NASA dataset

estimated intention by the Viterbi algorithm, which runs on cyclic explanation graphs. Figure 6.3 shows an example of an estimated plan, in which the purple node is the estimated intention, the green internal nodes are subgoals and the red leaf nodes represent actions and web pages accessed by the visitor. Parse trees like this can visualize the visitor's plan and help a web manager improve the website. For example, in this case, the actions taken from No. 4 to No. 6 are recognized as "Search"; therefore, the visitor's search may be aided by adding a link from the page of No. 3 to that of No. 7.

6.4.3 Consideration of experimental results

In the previous section, we experimentally compared the prefix method with existing methods using three probabilistic models: PCFG, logistic regression and HMM. This section presents a closer look at the results of this experiment.

First, we found that the PCFG method showed poor accuracy at prefix lengths below 10. This is thought to be caused by a mismatch between the model, which assumes the observed data is complete, and the incomplete data provided as input.

Second, the accuracy of the prefix method was higher than or equal to that of logistic regression, a standard discriminative model, when the prefix length was long. In contrast, logistic regression outperformed grammatical methods when prefix length was short. Grammatical methods require the correct identification of the most-likely parse tree for the top-intention; however, this identification becomes quite difficult for short sequences as they give little information on correct parse trees. This can result in mis-identification, which leads to poor performance.

Third, the accuracy of our method and the HMM method differed markedly. This difference may have been due to our use of universal session grammar to determine

correct answers in the experiment. The use of universal session grammar as a criterion of accuracy causes a substantial disadvantage to HMM, which is a special case of PCFG and not as expressive as universal session grammar.

Finally, the degree of difference in accuracy was dependent on the dataset. For example, the difference between the accuracy of the prefix and PCFG methods with that of the other methods was small in the U of S dataset, but was larger in the NASA dataset, especially for long prefix lengths (Figure 6.2). To understand this phenomenon, we computed the entropy of each PCFG model. The entropies of the U of S, ClarkNet and NASA datasets were 5.14×10^4, 2.77×10^5, and 3.14×10^6 respectively[7]. These calculations showed that the prefix and PCFG methods were more accurate than the other methods and that higher entropy of a data model in the experiment can co-occur. We do not think that this co-occurrence is accidental. First, because entropy is an indicator of the uncertainty of a probability distribution, entropy in PCFGs represents the uncertainty of parse trees. Thus, for simple data models and the low entropy, it is easy to determine correct intention, resulting in simple methods such as HMM and logistic regression being comparable to the prefix and PCFG methods. However when entropy is high, as in the NASA dataset, these simple approaches fail to identify the correct parse tree. Therefore, the prefix and PCFG methods, which that exploit structural information in the input data, outperform HMM and logistic regression, particularly when the input data is long.

6.5 Conclusion of this chapter

This chapter presents a new plan and intention recognition method, based on prefix probability computations via parse trees in PCFGs. This method can identify users' goals and plans based on their incomplete action sequences. Using three actual datasets, the prefix and infix methods introduced in this chapter were compared with several other methods of identifying visitors' intentions at a website, the PCFG method that always treats action sequences as complete sentences, the HMM method that uses a mixture of HMMs instead of a mixture of PCFGs, and logistic regression. The results empirically demonstrated the superiority of our approach for long (but incomplete) action sequences.

[7]Entropy is defined as $-\sum_\tau P(\tau) \log P(\tau)$, where τ is a possible parse tree [89]. In our setting, a common grammar, the universal session grammar, was used for all data sets. Therefore, entropy depends only on the parameters of a PCFG learned from the data set.

Chapter 7

Conclusion

7.1 Contribution

Realization of a PIE framework should include answers to the following research issues:

Issue 1. How to extract attributes of an event.

Issue 2. How to reconstruct scene structure.

To solve these two issues, we utilized two approaches to each and constructed the PIE framework.

- Approach 1. Information integration

  (1-a) Multi-sensory approach

  (1-b) Multi-attribute approach

- Approach 2. Introducing plan and intention recognition

  (2-a) Acquisition of background knowledge

  (2-b) Consideration of the incompleteness of observation

To evaluate these approaches, we investigated applications in Chapter 3-5 and propose the following methods for each approach:

Chapter 4 . Cooking recognition

(1-a) An audio-visual multimodal CNN.

(2-a) Construction of a recipe model from recipes on the web.

Chapter 5 . Bird song analysis

(1-b) A spatial-cue-based probabilistic model (SCBPM).

Chapter 6 . Web session log analysis

(2-b) A prefix method for plan and intention recognition.

The main contributions of these are summarized below:

7.1.1 Towards a general framework

Chapter 3 proposes a three layered framework for scene analysis, called the PIE framework. These layers correspond to the three tasks: event extraction considering information integration, intention recognition, and plan recognition. Separation of these three layers allowed them to be addressed individually. The first example consisted of cooking recognition (Chapter 4), followed by separately addressing the event layer (Chapter 5) and the intention and plan layers (Chapter 6).

7.1.2 Towards information integration

Information integration in scene analysis was addressed using two approaches, a multi-sensory and a multi-attribute approach. The first approach was applied to cooking event extraction in cooking recognition (Chapter 4). That experiment showed the audio-visual multimodal cooking event extraction effectively improved the occlusion-robustness of only-vision-based event extraction. The second approach was applied to bird song identification (Chapter 5). To address the co-dependency among sound sources, a probabilistic mode, SCBPM, was utilized to represent locations and bird classes. That experiment showed that the SCBPM outperformed the conventional GMM method in classifying separated sounds. This method will not only contribute to the progress of bird research but to research on robot audition, such as the processing separated sounds. Furthermore, these information integration methods will likely be applied to other areas.

7.1.3 Introducing plan and intention recognition

To consider the relationship among events, we propose utilizing plan and intention recognition techniques in which background knowledge will be provided as a plan model. This dissertation addressed two problems with this : 1) how to construct a plan mode representing the background knowledge and 2) how to deal with incomplete data. The first problem was encountered in cooking recognition (Chapter 4) and was addressed using recipes on web sites. That experiment showed that consideration of the top-n rather than the top-1 recipe was important in maintaining the accuracy required for applications. Chapter 6 addressed the second problem in web session log analysis. Plan recognition under conditions of incomplete data, as in online situations, required the introduction of prefix computations of PCFGs. Our method experimentally outperformed other, more conventional methods in that it was more accurate in the intention recognition task from incomplete data. These results suggest that plan and intention recognition can be expanded to address additional applications.

7.2 Remaining issues and future work

In this section, we discuss remaining issues for practical applications of scene analysis and future work. These remaining issues include:

Sensors and devices

Sensors and recording devices are important for scene analysis. This dissertation considered only a microphone (array), a camera, and digital logs like web access logs. However, additional other sensors would provide additional information about a scene. For example, bird song analysis (Chapter 5) used only a table-top microphone array. Our research group is now developing a microphone array to use in fields. This type of microphone array must be water resist for long-term use. We believe that such devices would result in more accurate recording in the field.

Overall optimization

The PIE framework should be optimized by considering the overall system. Because the PIE framework provides individual designs for each layer, parameters in these layers can

be optimized individually, but not as an overall framework. Because overall optimization cannot be achieved, future work should include overall optimization of this framework.

Application-specific issues

As the PIE framework is a general framework, it cannot resolve application-specific issues; e.g., preprocessing of data, methods of visualizing experiment results, and parameter tuning. Although this dissertation addressed these issues for three applications, these methods may not be directly applicable to other applications.

7.3 Conclusion

This dissertation dealt with a framework for scene analysis and development of a scene analysis system. We addressed two issues required for this type of scene analysis system:

Issue 1. How to extract attributes of an event.

Issue 2. How to reconstruct scene structure.

Chapter 1 addressed the background, issues and our approaches, as well as the organization of the dissertation. That chapter explained the importance of scene analysis and the goal of this study.

Chapter 2 presented a literature review of previous work related to scene analysis. It first described research related to visual and auditory processing, followed by a description of fundamental techniques in scene analysis, including probabilistic models with the SCBPM and plan/intention recognition was described. Finally, it addressed the relationship of this study with related work, clarifying that this study was based on existing technologies and was extended to more realistic problems.

Chapter 3 described the PIE framework and an example of this framework. It first described the importance and role of a framework for scene analysis systems, followed by the introduction of a cooking recognition system based on the PIE framework, to gain a better understanding of this PIE framework.

Chapters 4-6 report applications of scene analysis considering the issues described above.

Chapter 4 explains cooking recognition as an application of the PIE framework. Tasks in this application included **(1-a)** audio-visual event recognition using CNN and

(2-a) construction of HHMM to represent information of recipes from websites. The experimental results of event extraction, recipe recognition, and procedure recognition showed the effectiveness of these approaches. Based on this result, we built a prototype of a cooking support system.

Chapter 5 describes bird song analysis, focusing on the event layer in the PIE framework and audio data recorded with a microphone array. This chapter addressed the problem of co-dependency among separated sounds. This problem was resolved by applying the SCBPM to integrate robot audition techniques such as sound source localization, separation, and identification **(1-b)**. The SCBPM was experimentally evaluated using a prototype semi-automatic annotation system. Experimental results showed that the system with the SCBPM outperformed the conventional system using GMM, as shown by accuracy of bird song identification.

Chapter 6 utilized web session log analysis to address plan and intention recognition. A new method was proposed to deal with incomplete data **(2-b)** in a realistic situation. This method was based on prefix probability computation and was shown effectiveness experimentally using real web session log datasets.

Chapter 7 described the main contributions of this study and the applicability of the PIE framework. The result of this study can contribute to information integration and to the introduction of plan and intention recognition into scene analysis. We also discussed remaining issues and future work.

Using the PIE framework, we were able to expand scene analysis techniques to more realistic settings. We hope that our study will trigger further attempts to develop scene analysis systems. Furthermore, we believe that our applications may be good examples of further application to scene analysis.

Author's publications

Chapters 3, 4
Scene Analysis Framework and Its Application to Cooking Recognition

- R.Kojima, O.Sugiyama, K.Nakadai: Multimodal scene understanding framework and its application to cooking recognition. Applied Artificial Intelligence, Vol. 30 No. 3 pp. 181-200, 2016.

- R.Kojima, O.Sugiyama, K.Nakadai: Audio-visual scene understanding utilizing text information for a cooking support robot. Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference, September, 2015.

- R.Kojima, O.Sugiyama, K.Nakadai: Scene understanding based on sound and text information for a cooking support robot. Current Approaches in Applied Artificial Intelligence: 28th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2015, Seoul, South Korea, June, 2015.

- R.Kojima, O.Sugiyama, K.Nakadai: Multimodal scene understanding using CNN and hierarchical HMM for a cooking support robot. Machine Learning Summer School, 2015.

- 小島 諒介, 杉山 治, 中臺 一博: 調理認識を対象としたレシピの確率モデルとその学習. 第 15 回計測自動制御学会システムインテグレーション部門講演会, Dec., 2014.

- 小島 諒介, 杉山 治, 中臺 一博: 調理支援のための音情報を用いた料理レシピの推定. 第 32 回日本ロボット学会学術講演会（RSJ2014）, Sep., 2014.

Chapter 5
Bird Song Analysis

- R.Kojima, O.Sugiyama, K.Hoshiba, K.Nakadai, R.Suzuki, C. E.Taylor: Bird song scene analysis using a spatial-cue-based probabilistic model. Journal of Robotics and Mechatronics, Vol. 29 No. 1 , 2017(accepted).

- R.Kojima, O.Sugiyama, R.Suzuki, K.Nakadai, C. E.Taylor: Semi-Automatic Bird Song Analysis by Spatial-Cue-Based Integration of Sound Source Detection, Localization, Separation, and Identification. Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference, October, 2016.

- 小島 諒介, 杉山 治, 干場 功太郎, 鈴木 麗璽, 中臺 一博: 空間情報を用いた鳥の歌分析. 人工知能学会 第 46 回 AI-Challenge 研究会, Nov., 2016.

- 小島 諒介, 杉山 治, 鈴木 麗璽, 中臺 一博: 音源位置を考慮した音源同定のための確率モデルとその学習. 第 33 回日本ロボット学会学術講演会 (RSJ2016), Sep., 2016.

- 小島 諒介, 杉山 治, 鈴木 麗璽, 中臺 一博: 音源アノテーション補助のための音源位置を考慮した同定モデル. 第 33 回日本ロボット学会学術講演会 (RSJ2015), Sep., 2015.

Chapter 7
Web Session Log Analysis

- R.Kojima, T.Sato: Goal and plan recognition via parse trees using prefix and infix probability computation. Inductive Logic Programming, LNAI 9046, pp. 76-91, 2015.

- 小島 諒介, 佐藤 泰介: アクセスログ分析における接頭部分列からのプラン認識. In 人工知能学会論文誌, Vol. 29 No. 3 pp. 301-310, 2014.

- R.Kojima, T.Sato: Prefix and infix probability computation in PRISM. Probabilistic Logic Programming 2014, An ICLP workshop, 2014.

- R.Kojima, T.Sato: Goal recognition from incomplete action sequences by probabilistic grammars. the 24th International Conference on Inductive Logic Programming(short paper), September, 2014.
- 小島諒介, 佐藤泰介: Prefix 確率を用いたプラン認識の Web アクセスログ解析への応用. 第 3 回 CSPSAT2 研究会 (2013), 2013.

Author's Publications Not Cited in This Thesis

Journal Article

- O.Sugiyama, S.Uemura, A.Nagamine, R.Kojima, K.Nakamura, K.Nakadai: Outdoor acoustic event identification with DNN using a quadrotor-embedded microphone array Journal of Robotics and Mechatronics, Vol. 29 No. 1 , 2017(accepted).

- K.Hoshiba, O.Sugiyama, A.Nagamine, R.Kojima, M.Kumon, K.Nakadai: Design and assessment of sound source localization system with a UAV-embedded microphone array. Journal of Robotics and Mechatronics, Vol. 29 No. 1 , 2017(accepted).

- T.Ohata, K.Nakamura, A.Nagamine, T.Mizumoto, T.Ishizaki, R.Kojima, O.Sugiyama, K.Nakadai: Outdoor sound source detection for a quadrotor with a microphone array. Journal of Robotics and Mechatronics, Vol. 29 No. 1 , 2017(accepted).

- S.Matsubayashi, R.Suzuki, F.Saito, T.Murate, T.Masuda, K.Yamamoto, R.Kojima, K.Nakadai, H. G.Okuno: Acoustic monitoring of the great reed warbler using multiple microphone arrays and robot audition. Journal of Robotics and Mechatronics, Vol. 29 No. 1 , 2017(accepted).

Papers Presented at International Conferences

- T.Morito, O.Sugiyama, R.Kojima, K.Nakadai: Partially shared deep neural network in sound source separation and identification using a UAV-embedded microphone array. Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference, 2016.

- T.Morito, O.Sugiyama, S.Uemura, R.Kojima, K.Nakadai: Reduction of computational cost using two-stage deep neural network for training for denoising and sound source identification. Trends in Applied Knowledge-Based Systems and Data Science - 29th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2016, Morioka, Japan, August, 2016.

- S.Uemura, O.Sugiyama, R.Kojima, K.Nakadai: Outdoor acoustic event identification using sound source separation and deep learning with a quadrotor-embedded microphone array. The 6th International Conference on Advanced Mechatronics (ICAM2015), December, 2015.
- O.Sugiyama, R.Kojima, K.Nakadai: Interactive interface to optimize sound source localization based on microphone array with coarse-to-fine tuning for humanoids. Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference, 2015.
- O.Sugiyama, R.Kojima, K.Nakadai: Interactive interface to optimize sound source localization with HARK. Current Approaches in Applied Artificial Intelligence: 28th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2015, Seoul, South Korea, June, 2015.

Domestic Conference

- 小島諒介, 亀谷由隆, 佐藤泰介: Naive Bayes モデルを用いた効率的なクラスタラベリング手法 (特集 「Big data と機械学習・データサイエンス」 および一般). In 人工知能基本問題研究会, Vol. 88 pp. 19-24, 2013.
- 小島 諒介, 渡邊 恵太, 稲見 昌彦, 五十嵐 健夫: VisualHaptics 2.0: 感触の解像度向上とタッチパネルへの応用. In インタラクション, 2013.
- 奥乃 博公文 誠杉山 治糸山 克寿 小島 諒介 干場 功太郎水本 武志坂東 宜昭中臺 一博: HARK 2.3 の紹介とタフロボティクスチャレンジへの展開. 第 17 回計測自動制御学会システムインテグレーション部門講演会 (SI2016), Dec., 2016.
- 干場 功太郎, 小島 諒介, 杉山 治, 中臺 一博: UAV 搭載マイクロホンアレイを用いた音源探査の実環境評価. 第 33 回日本ロボット学会学術講演会 (RSJ2016), Sep., 2016.
- 杉山 治, 小島 諒介, 中臺 一博: UAV 搭載マイクアレイを用いた高雑音環境下における音イベント検出・識別の並列最適化. 人工知能学会 第 46 回 AI-Challenge 研究会, Nov., 2016.
- 杉山 治, 上村 知史, 小島 諒介, 中臺 一博: UAV 搭載マイクアレイを用いた高雑音環境下における音イベント検出・識別システム. ロボティクス・メカトロニクス講演会 ROBOMECH2016, Jun., 2016.

- 杉山 治, 小島 諒介, 中臺 一博: Coarse-to-Fine チューニングを用いた HARK の音源定位パラメータの最適化. 人工知能学会 第 43 回 AI-Challenge 研究会, Nov., 2015.

- 杉山 治, 小島 諒介, 中臺 一博: Coarse-to-fine チューニングに基づく HARK を用いた 音源定位パラメータ最適化インタフェースの提案. 第 33 回日本ロボット学会学術講演会 (RSJ2015), Sep., 2015.

- 杉山 治, 小島 諒介, 中臺 一博: ユーザとの対話に基づきブログを自動生成するブログロボットの提案. 第 15 回計測自動制御学会システムインテグレーション部門講演会, Dec., 2014.

- 杉山 治, 小島 諒介, 中臺 一博: 音環境理解ロボットを対象とした環境音識別器のための協調型インタフェースの提案. 第 32 回日本ロボット学会学術講演会 (RSJ2014), Sep., 2014.

- 松林 志保, 小島 諒介, 中臺 一博, 鈴木 麗璽: 複数のマイクロフォンアレイで測る野鳥の位置. 第 10 回バードリサーチ大会, Nov., 2015.

- 松林 志保, 鈴木 麗璽, 小島 諒介, 中臺 一博: 複数のマイクロフォンアレイとロボット聴覚ソフトウェア HARK を用いた野鳥の位置観測精度の検討. 人工知能学会 第 43 回 AI-Challenge 研究会, Nov., 2015.

- 松林 志保, 小島 諒介, 中臺 一博, 鈴木 麗璽: 複数のマイクロホンアレイによる野鳥の位置観測精度の検討. 日本鳥学会 2015 年度大会, Sep., 2015.

- 森戸 隆之, 杉山 治, 小島 諒介, 中臺 一博: 部分共有アーキテクチャを用いた深層学習ベースの音源同定の検討. 人工知能学会 第 46 回 AI-Challenge 研究会, Nov., 2016.

- 森戸 隆之, 杉山 治, 小島 諒介, 中臺 一博: 部分共有型 Deep Neural Network を用いた音源同定. ロボティクス・メカトロニクス講演会 ROBOMECH2016, June, 2016.

- 森戸 隆之, 杉山 治, 上村 知史, 小島 諒介, 中臺 一博: 深層学習による多チャネル音響信号に対する音源同定の検討. 第 16 回 公益社団法人 計測自動制御学会 システムインテグレーション部門 講演会 SI2015, Dec., 2015.

- 上村 知史, 杉山 治, 小島 諒介, 中臺 一博: UAV を用いた屋外音環境理解における音源検出・識別の評価. 第 16 回 公益社団法人 計測自動制御学会 システムインテグレーション部門 講演会 SI2015, Dec., 2015.

- 上村 知史, 杉山 治, 小島 諒介, 中臺 一博: クアドロコプタ搭載マイクロホンアレイを用いた 音源分離と深層学習による音源識別. 第 33 回日本ロボット学会学術講演会 (RSJ2015), Sep., 2015.

- 上村 知史, 杉山 治, 小島 諒介, 大畑 琢磨, 中臺 一博: クアドロコプタ搭載マイクロホンアレイを用いた深層学習による音声識別. 第 15 回計測自動制御学会システムインテグレーション部門講演会, Dec., 2014.

- 長峰 諒英, 大畑 琢磨, 上村 知史, 小島 諒介, 杉山 治, 中村 圭佑, 中臺 一博: 屋外音環境理解における音源検出の性能評価と可視化. 第 41 回人工知能学会 AI チャレンジ研究会, Nov., 2014.

Domestic Symposium

- 小島 諒介: マイクロフォンアレイを利用した確率的環境理解フレームワーク. 基盤研究（S）「ロボット聴覚の実環境理解に向けた多面的展開」最終シンポジウム, Dec., 2016.

Graduation Thesis

- 小島 諒介: Naive Bayes モデルに基づき得られたクラスタの AND/OR 式によるラベル付け，東京工業大学工学部情報工学科，卒業論文，2012.

- 小島 諒介: プラン認識による系列データの解析，東京工業大学情報理工学研究科計算工学専攻，修士論文，2014.

Patent

- 小島 諒介，中臺一博: “環境理解装置および環境理解方法” 特開 2016-51052

Awards

- 2016 第 33 回日本ロボット学会研究奨励賞

- 2016 人工知能学会研究会 第 43 回 AI チャレンジ研究会優秀賞
- 2014 SICE SI 部門講演会 優秀講演者賞
- 2014 ILP 2014 Best Student Paper runner-up

Appendix

This appendix describes the methods used to compute the probability of the *prefix* and *infix* sentences (Chapter 6) in PRISM, a method that can represent many types of probabilistic distributions using *distribution semantics* [90, 85] . This system was used to implement HHMM and PCFG in Chapter 4 and Chapter 6, respectively. Furthermore, by implementing more general prefix computations (called *computation on the cyclic explanation graph*) than that of PCFG, it became available as open source software[1]. This appendix first reviews PRISM briefly, followed by explanations of the prefix and infix probability computations in PRISM. Further details about the PRISM system and other computation mechanisms are provided by the PRISM document[1]. We hope that our study will trigger further attempts to develop a plan and intention recognition system and the other reasoning systems.

A.1 Reviewing PRISM

PRISM is a probabilistic extension of Prolog. It supports general and highly efficient computations of probability and parameter learning for a wide variety of probabilistic models including PCFGs. For example a PRISM program for PCFGs can perform probability computation in the same time complexity as the Inside-Outside algorithm, an algorithm specialized for computing probabilities for PCFGs. In PRISM, probabilities are computed by dynamic programming applied to *acyclic explanation graph*s, which are internal data structures encoding all (but finitely many) proof trees.

PRISM provides a basic built-in predicate in the form `msw`(i, v), representing a probabilistic choice used in probabilistic modeling. This predicate is called a "multi-valued random switch" (`msw`) and is used to denote a simple probabilistic event $X_i = v$ where X_i is a discrete random variable and v is its realized value. $V_i = \{v_1, \cdots, v_{|V_i|}\}$ is the

[1] PRISM2.2: `http://rjida.meijo-u.ac.jp/prism/`

set of possible outcomes of X_i, and i is the *switch name* of $\mathtt{msw}(i, v)$.

To represent the distribution $P(X_i = v)$ $(v \in V_i)$, a set $\{\mathtt{msw}(i, v) \mid v \in V_i\}$ of mutually exclusive `msw` atoms is introduced, with a joint distribution such that $P(\mathtt{msw}(i, v)) = \theta_{i,v} = P(X_i = v)$ $(v \in V_i)$ where $\sum_{v \in V_i} \theta_{i,v} = 1$. $\{\theta_{i,v}\}$ are called *parameter*s, and the set of all parameters $\mathbf{\Theta}$ appearing in a program is manually specified by the user or automatically learned from data.

Suppose a positive program DB, which is a Prolog program containing `msw` atoms. A *basic distribution* (probability measure) $P_{\mathtt{msw}}(\cdot \mid \mathbf{\Theta})$ is defined as the product of distributions for the `msw`s appearing in DB. The basic distribution can be uniquely extended by way of the least model semantics in logic programming to a σ-additive probability measure $P_{DB}(\cdot \mid \mathbf{\Theta})$ over possible Herbrand interpretations of DB. The denotation DB in *distribution semantics* [90, 85] is standard semantics for probabilistic logic programming. In the following, $\mathbf{\Theta}$ is omitted when the context is clear.

Semantically PRISM is one of many possible implementations of the distribution semantics that efficiently computes probability by adding two assumptions, *independence* and *exclusiveness* . Let G be a non-msw atom which is ground. $P_{DB}(G)$, the probability of G defined by the program DB, can be naively computed as follows. First reduce the top-goal G using Prolog's exhaustive top-down proof search to a propositional DNF(disjunctive normal form) formula $\mathrm{expl}_0(G) = e_1 \vee e_2 \vee \cdots \vee e_k$ where $e_i (1 \leq i \leq k)$ is a conjunction of atoms $\mathtt{msw}_1 \wedge \cdots \wedge \mathtt{msw}_n$ such that $e_i, DB \vdash G$[2]. Each e_i is called an *explanation* for G. Then assuming the

Independence condition (`msw` atoms in an explanation are independent):

$$P_{DB}(\mathtt{msw} \wedge \mathtt{msw}') = P_{DB}(\mathtt{msw}) P_{DB}(\mathtt{msw}')$$

Exclusive condition (explanations are exclusive):

$$P_{DB}(e_i \wedge e_j) = 0 \quad \text{if} \quad i \neq j$$

we compute $P_{DB}(G)$ as

[2] $\mathrm{expl}_0(G)$ is equivalent to G in view of the distribution semantics. When convenient, we treat $\mathrm{expl}_0(G)$ as a bag $\{e_1, e_2, \cdots, e_k\}$ of explanations.

$$
\begin{aligned}
P_{DB}(G) &= P_{DB}(e_1) + \cdots + P_{DB}(e_k) \\
P_{DB}(e_i) &= P_{DB}(\mathtt{msw}_1) \cdots P_{DB}(\mathtt{msw}_n) \quad \text{for} \quad e_i = \mathtt{msw}_1 \wedge \cdots \wedge \mathtt{msw}_n
\end{aligned}
$$

Recall that `msw`s with different switch names are independent by construction of $P_{\mathtt{msw}}(\cdot \mid \mathbf{\Theta})$. We further assume that `msw` atoms with the same switch name are *iid* (independent and identically distributed). Fortunately this assumption can be automatically satisfied.

Contrastingly the exclusiveness condition cannot be automatically satisfied. It needs to be satisfied by the user, for example, by writing a program so that it generates an output solely as a sequence of probabilistic choices made by `msw` atoms (modulo auxiliary non-probabilistic computation). Although most generative models including BNs, HMMs and PCFGs are naturally written in this style, there are models which are not [91]. Relating to this, observe that *Viterbi explanation*, i.e., the most likely explanation e^* for G, is computed similarly to $P_{DB}(G)$ just by replacing sum with argmax: $e^* \stackrel{\text{def}}{=} \underset{e \in \mathrm{expl}_0(G)}{\mathrm{argmax}}\ P_{DB}(e)$.

So far our computation is naive. Since there can be exponentially many explanations, naive computation would lead to exponential time computation. PRISM avoids this by adopting *tabled search* in the exhaustive search for all explanations for the top-goal G and applying dynamic programming to probability computation. By tabling, a goal which is once called and proved is stored (tabled) in memory with its answer substitutions and later calls to the same goal return with a stored answer substitution without processing further. Tabling is important to probability computation because tabled goals factor out common sub-conjunctions in $\mathrm{expl}_0(G)$, which results in sharing probability computation for common sub-conjunctions, thereby realizing dynamic programming which gives exponentially faster probability computation compared to naive computation.

As a result of exhaustive tabled search for all explanations for G, PRISM yields a set of propositional formulas called *defining formula*s of the form $H \Leftrightarrow B_1 \vee \cdots \vee B_h$ for every tabled goal H that directly or indirectly calls `msw`s. We call the heads of defining formulas *defined goal*s. Each $B_i (1 \leq i \leq h)$ is recursively composed of a conjunction $C_1 \wedge \cdots \wedge C_m \wedge \mathtt{msw}_1 \wedge \cdots \wedge \mathtt{msw}_n (0 \leq m, n)$ of defined goals $\{C_1, \cdots, C_m\}$ and `msw` atoms $\{\mathtt{msw}_1, \cdots, \mathtt{msw}_n\}$. We introduce a binary relation $H \succ C$ over defined goals such that

$H \succ C$ holds if H is the head of some defining formula and C occurs in the body. We denote by expl(G) the whole set of defining formulas and call expl(G) the *explanation graph* for G as in the non-tabled case. When "$\succ$" is acyclic, we call an explanation graph *acyclic* and extend "$\succ$" to a partial ordering over the defined goals.

Once expl(G) is obtained as an acyclic explanation graph, since defined goals are layered by the "$\succ$" relation, defining formulas in the bottom layer (minimal elements) have only `msw`s in their bodies whose probabilities are known (declared in the program), so we can compute probabilities by a sum-product operation for all defined goals from the bottom layer upward in a dynamic programming manner in time linear in the number of atoms appearing in expl(G).

Compared to naive computation, the use of dynamic programming on expl(G) can reduce time complexity for probability computation from exponential time to polynomial time. For example PRISM's probability computation for HMMs takes $O(L)$ time for a given sequence with length L and coincides with the standard forward-backward algorithm for HMMs. Likewise PRISM's sentence probability computation for PCFGs takes $O(L^3)$ time for a given sentence with length L and coincides with inside probability computation for PCFGs.

Viterbi inference that computes the Viterbi explanation and its probability is similarly performed on expl(G) in a bottom-up manner like probability computation stated above. The only difference is that we use argmax instead of sum. In what follows, we look into the detail of how the Viterbi explanation is computed.

Let H be a defined goal and $H \Leftrightarrow B_1 \vee \cdots \vee B_h$ the defining formula for H in expl(G). Write $B_i = C_1 \wedge \cdots \wedge C_m \wedge \mathtt{msw}_1 \wedge \cdots \wedge \mathtt{msw}_n (0 \leq m, n)$ $(1 \leq i \leq h)$ and suppose recursively that the Viterbi explanation $e^*_{C_j} (1 \leq j \leq m)$ has already been calculated for each defined goal in C_j in B_i. Then the Viterbi explanation $e^*_{B_i}$ for B_i and Viterbi explanation e^*_H are respectively computed by

$$
\begin{aligned}
e^*_{B_i} &= e^*_{C_1} \wedge \cdots \wedge e^*_{C_m} \wedge \mathtt{msw}_1 \wedge \cdots \wedge \mathtt{msw}_n \\
e^*_H &= \operatorname*{argmax}_{B_i} P_{DB}(e^*_{B_i} \mid \boldsymbol{\Theta}) \qquad (1) \\
&\qquad \text{where } P_{DB}(e^*_{B_i} \mid \boldsymbol{\Theta}) = P_{DB}(e^*_{C_1}) \cdots P_{DB}(e^*_{C_m}) \theta_{i_1,v_1} \cdots \theta_{i_n,v_n}
\end{aligned}
$$

Here θ_{i_1,v_1} is a parameter associated with $\mathtt{msw}_1$ and so on. In this way, the Viterbi explanation for the top-goal G is computed in a bottom-up manner by scanning expl(G) once in time linear in the size of expl(G) in an acyclic explanation graph.

```
values(s,[[s,s],[a],[b]]).
:- set_sw(s,[0.4,0.3,0.3]).

pre_pcfg(L):- pre_pcfg([s],L,[]).        % (1) L is a prefix
pre_pcfg([A|R],L0,L2):-                  % (2) L0 is ground when called
( get_values(A,_) -> msw(A,RHS),      % (3) if A is a nonterminal
pre_pcfg(RHS,L0,L1)                % (4) select rule A->RHS
;  L0=[A|L1] ),                       % (5) else consume A in L0
( L1=[] -> L2=[]                      % (6) (pseudo) success
; pre_pcfg(R,L1,L2) ).                % (7) recursion
pre_pcfg([],L1,L1).                      % (8) termination
```

Figure 1: Prefix parser DB_0

A.2 Prefix computation in PRISM

In this section, we examine prefix computation for PCFGs in PRISM. A PCFG $\mathsf{G}_{\mathbf{\Phi}}$ is a CFG G augmented with a parameter set $\mathbf{\Phi} = \bigcup_{N^i \in \mathbf{N}} \{\phi_r\}_{N^i}$ where $\mathbf{N}$ is a set of nonterminals, N^1 a start symbol and $\{\phi_r\}_{N^i}$ the set of parameters associated with rules $\{r \mid r = N^i \to \zeta\}$ for a nonterminal N^i where ζ is a sequence of nonterminal and terminal symbols. We assume that the ϕ_r's satisfy $0 < \phi_r < 1$ and $\sum_{\zeta: N^i \to \zeta} \phi_{N^i \to \zeta} = 1$.

There are already algorithms to compute prefix probabilities in PCFGs [83, 84]. We here briefly describe prefix probability computation based on explanation graphs in PRISM [92]. As previously stated, a *prefix* $\mathbf{v}$ is an initial substring of a sentence and the prefix probability $P^{N^1}_{\mathrm{pre}}(\mathbf{v})$ of $\mathbf{v}$ is an infinite sum of probabilities of sentences extending $\mathbf{v}$:

$$P^{N^1}_{\mathrm{pre}}(\mathbf{v}) = \sum_{\mathbf{w}} P_{\mathsf{G}}(\mathbf{vw})$$

where $\mathbf{w}$ ranges over strings such that $\mathbf{vw}$ is a sentence in G. Prefix probabilities are computed in PRISM by way of cyclic explanation graphs. We sketch our prefix probability computation following [92]. We use a PCFG $\mathsf{G}_0 = \{\ \mathtt{s} \to \mathtt{s}\,\mathtt{s} : 0.4,\ \mathtt{s} \to \mathtt{a} : 0.3,\ \mathtt{s} \to \mathtt{b} : 0.3\ \}$ where "$\mathtt{s}$" is a start symbol and "$\mathtt{a}$" and "$\mathtt{b}$" are terminals and consider the computation of the prefix probability $P^{\mathtt{s}}_{\mathrm{pre}}(\mathtt{a})$ of prefix $\mathtt{a}$. To compute $P^{\mathtt{s}}_{\mathrm{pre}}(\mathtt{a})$, we first parse "$\mathtt{a}$" as a prefix by the PRISM program DB_0 in Fig. 1. As can be seen from the

`pre_pcfg([a]) <=> pre_pcfg([s],[a],[])`	:	$\mathrm{P}(\texttt{pre\_pcfg([a])}) = \mathtt{X} = \mathtt{Y}$
`pre_pcfg([s],[a],[])`	:	$\mathrm{P}(\texttt{pre\_pcfg([s],[a],[])}) = \mathtt{Y}$
`<=> pre_pcfg([s,s],[a],[]) & msw(s,[s,s])`		$= \mathtt{Z} \cdot \theta_{\mathtt{s}\to\mathtt{ss}} + \mathtt{W} \cdot \theta_{\mathtt{s}\to\mathtt{a}}$
`v pre_pcfg([a],[a],[]) & msw(s,[a])`		
`pre_pcfg([s,s],[a],[])`	:	$\mathrm{P}(\texttt{pre\_pcfg([s,s],[a],[])}) = \mathtt{Z}$
`<=> pre_pcfg([a],[a],[]) & msw(s,[a])`		$= \mathtt{W} \cdot \theta_{\mathtt{s}\to\mathtt{a}} + \mathtt{Z} \cdot \theta_{\mathtt{s}\to\mathtt{ss}}$
`v pre_pcfg([s,s],[a],[]) & msw(s,[s,s])`		
`pre_pcfg([a],[a],[])`	:	$\mathrm{P}(\texttt{pre\_pcfg([a],[a],[])}) = \mathtt{W} = 1$

Figure 2: Explanation graph for prefix "a" (left) and associated probability equations (right)

comments, it runs exactly like a standard top-down CFG parser except *pseudo success* at line `(6)`. *pseudo success* means an immediate return with success on the consumption of the input prefix `L1` ignoring the remaining nonterminals in `R` at line `(2)`[3].

The Viterbi algorithm in Eq.(1) for acyclic explanation graphs is no longer applicable to cyclic graphs as it wouldn't stop if applied to them. So we generalize it for cyclic explanation graphs using a shortest path algorithm such as Dijkstra's algorithm and the Bellman-Ford algorithm [94]. In our implementation, we adopted the Bellman-Ford algorithm since it neither requires additional data structure nor memory by reusing the space for the Viterbi algorithm.

By running a command `?-probf(pre_pcfg([a]))` in PRISM, we obtain an explanation graph in Figure 2 (left) for `pre_pcfg([a])`[4]. Note a cycle exists in the explanation graph; `pre_pcfg([s,s],[a],[])` calls itself in the third defining formula. Since this is a small example, its explanation graph has only self-loops. In general however, an explanation graph for prefix parsing has larger cycles as well as self-loops, and we call this type of explanation graphs *cyclic explanation graphs* [92].

Then we convert the defining formulas to a set of probability equations about `X`, `Y`, `Z` and `W` as shown in Figure 2 (right). We use the assumptions in PRISM that goals are independent ($P(A \wedge B) = P(A)P(B)$) and disjunctions are exclusive ($P(A \vee B) = P(A)+P(B)$). By solving them using parameter values $\theta_{\mathtt{s}\to\mathtt{ss}} = 0.4$ and $\theta_{\mathtt{s}\to\mathtt{a}} = 0.3$ set by

[3]This is justified because we assume the consistency of PCFGs [93] that implies the probability of remaining nonterminals in `R` yielding some terminal sequences is 1.

[4]`probf/1` is a PRISM's built-in predicate and displays an explanation graph.

`:-set_sw(s,[0.4,0.3,0.3])` in the program DB_0, we finally obtain $\mathtt{X} = \mathtt{Y} = \mathtt{Z} = 0.5$[5]. So we have $P^{\mathtt{s}}_{\mathrm{pre}}(\mathtt{a}) = \mathtt{X} = 0.5$. In general, a set of probability equations generated from a prefix in a PCFG using DB_0 is always linear and solvable by matrix operation [92].

We next describe an extension of the Viterbi inference of PRISM to cyclic explanation graphs. The most likely explanation and its probability for cyclic explanation graphs is defined as usual as $e^* \stackrel{\mathrm{def}}{=} \underset{e \in \mathrm{expl}_0(G)}{\mathrm{argmax}}\, P_{DB}(e)$ where $\mathrm{expl}_0(G)$ is possibly an infinite set of explanations represented by a cyclic explanation graph. For example, the set of explanations represented by Figure 2 (left) is
$\mathrm{expl}_0(\mathtt{pre\_pcfg([s,s],[a],[])}) = \{$ `msw(s,[a])`, `msw(s,[s,s])` $\wedge$ `msw(s,[a])`, `msw(s,[s,s])` $\wedge$ `msw(s,[s,s])` $\wedge$ `msw(s,[a])`, $\cdots\}$ where the repetition of `msw(s,[s,s])` is produced by the cycle. Note that although there are infinitely many explanations, the most likely explanation is `msw(s,[a])` since the product of probabilities is monotonically decreasing w.r.t. the number of occurrences of `msw(s,[s,s])` ($0 < P_{DB}(\mathtt{msw}(\mathtt{s},[\mathtt{s},\mathtt{s}])) < 1$) in an explanation.

A.3 Infix probability computation in PRISM

Infix probability computation: beyond prefix probability computation

Up until now we have only considered prefixes that describe session logs under online situation or unachieved visitors. When we consider the cross-site situation however, infix needs to be introduced. Compared to prefix probability computation, infix probability computation is much harder and early attempts put some restrictions on it. However Nederhof and Satta recently proposed a completely general method to compute infix probability that solves a set of non-linear equations [95]. One thing to recall is that their method is purely numerical and yields no parse trees for prefixes or infixes, and hence cannot be used for Viterbi inference to infer the most likely parse tree for a given infix. Contrastingly our approach can yield parse trees for infixes as well as for prefixes.

[5] $\mathtt{W} = 1$ because `pre_pcfg([a],[a],[])` is logically proved without involving `msw`s.

Nederhof and Satta's algorithm

An *infix* $\mathbf{v}$ in a PCFG G is a substring of a sentence written as $\mathbf{uvw}$ for some terminal sequences $\mathbf{u}$ and $\mathbf{w}$. The infix probability $P_{\text{in}}^{N^1}(\mathbf{v})$ is defined as

$$P_{\text{in}}^{N^1}(\mathbf{v}) = \sum_{\mathbf{u},\mathbf{w}} P_{\mathsf{G}}(\mathbf{uvw})$$

where $\mathbf{u}$ and $\mathbf{w}$ range over strings such that $\mathbf{uvw}$ is a sentence. According to Nederhof and Satta [95], $P_{\text{in}}^{N^1}(\mathbf{v})$ is computed by first constructing an intersection PCFG $\mathsf{G}' = \mathsf{G} \cap \mathsf{FA}$ of G and a finite automaton FA which accepts every string containing $\mathbf{v}$, and second by computing the sum of probabilities of all sentences derived from G'. The second computation is reduced to solving a set of multi-variate polynomial equations (details omitted).

The problem here is that while their algorithm is completely general, building the intersection PCFG G' contains redundancy. Let $A \to BC$ be a CFG rule in G and $\{s_0, \ldots, s_n\}$ a set of states in FA. To create G', rules of the form $\langle s_i A s_k \rangle \to \langle s_i B s_j \rangle \langle s_j C s_k \rangle$ are constructed for every possible combination of states s_i, s_j, s_k ($0 \leq i, j, k \leq n$)[6] but many of these rules are not used to derive a sentence and need to be removed as useless rules.

Infix parsing and cyclic explanation graphs

To avoid building redundant rules by blindly combining states and removing them later, we here propose to introduce parsing to the Nederhof and Satta's algorithm. More concretely, we parse an infix `L` by the PRISM program in Fig. 3. It is a modification of the prefix parser in Fig. 1 that faithfully simulates the parsing action of the intersection PCFG G'.

This program differs from the prefix parser in that an input infix $\mathbf{w} = w_1 \cdots w_n$ is asserted in the memory as a sequence of state transitions: `tr(0,`w_1`,1)`, ..., `tr(`$n-1$`,`w_n`,`n`)`, together with other transitions constituting the finite automaton FA. In the program, `tr(S0,A,S1)` represents a state transition from `S0` to `S1` by a word `A` in the infix. `infix_pcfg(S0,S2,`α`)` reads that α, a sequence of terminals and nonterminals, spans a terminal sequence which causes a state transition of FA from `S0` to `S2`. Parsing an infix by the infix parser in Fig. 3 yields an explanation graph which is (mostly) cyclic

[6] This is to simulates a state transition of FA made by a string derived from the nonterminal A using $A \to BC$.

```
values(s,[[s,s],[a],[b]]).
:- set_sw(s,[0.4,0.3,0.3]).

infix_pcfg(L):-                    % L : input infix
build_FA(L),                     % FA asserted in the memory
assert_last_state(L,End),        % last_state(end(End)) asserted
start_symbol(C),
infix_pcfg(0,End,[C]).           % FA transits from state 0 to End
infix_pcfg(S0,S2,[A|R]):-
( get_values(A,_) ->             % A : nonterminal
msw(A,RHS),                   % use A -> RHS to expand A
infix_pcfg(S0,S1,RHS)
; tr(S0,A,S1) ),                 % state transition by A from S0 to S1
( last_state(end(S1)) -> S2=S1  % pseudo success
; infix_pcfg(S1,S2,R) ).
infix_pcfg(S,S,[]).
```

Figure 3: Infix parser DB_2

and converted to a set of probability equations just like the case of prefix probability computation. Unlike prefix probability, though, probability equations for an infix are (usually) non-linear and we solve them by Broyden's method, a quasi-Newton method. In this way, we can compute infix probability by way of cyclic explanation graphs. In addition, this program produces the most likely infix parse trees by the Viterbi algorithm on cyclic explanation graphs as explained in Section 6.2.

We experimentally applied the infix method to web session log data and obtained similar results to Figure 6.2(details omitted). However a set of non-linear equations for infix probability computation has multiple roots and a solution given by Broyden's method depends critically on the initial value. Moreover, since Broyden's method is a general solver for non-linear equations, its solution is not necessarily constrained to be between 0 and 1 and actually it often invalid. How to obtain a valid and stable solution of non-linear equations for infix probability computation remains as an open problem.

References

[1] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 91–99.

[3] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1778–1785.

[4] A. Gupta and L. S. Davis, "Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers," in *Proceedings of the European Conference on Computer Vision*. Springer, 2008, pp. 16–29.

[5] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: a review," *ACM Computing Surveys*, vol. 43, no. 3, p. 16, 2011.

[6] S.-C. Zhu and D. Mumford, *A Stochastic Grammar of Images*. Now Publishers Inc, 2007.

[7] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis, "Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2012–2019, 2009.

[8] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[9] H. Nakajima, K. Nakadai, Y. Hasegawa, and H. Tsujino, "Correlation matrix estimation by an optimally controlled recursive average method and its application to blind source separation," *Acoustical Science and Technology*, vol. 31, no. 3, pp. 205–212, 2010.

[10] K. Nakamura, K. Nakadai, F. Asano, and G. Ince, "Intelligent sound source localization and its application to multimodal human tracking," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 143–148.

[11] K. Nakamura, K. Nakadai, and G. Ince, "Real-time super-resolution sound source localization for robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 694–699.

[12] S. Uemura, O. Sugiyama, R. Kojima, and K. Nakadai, "Outdoor acoustic event identification using sound source separation and deep learning with a quadrotor-embedded microphone array," in *ICAM2015*. JSME, 2015, pp. 329–330.

[13] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis*. John Wiley & Sons, 2004, vol. 46.

[14] H. Sawada, R. Mukai, S. Araki, and S. Makino, "A robust and precise method for solving the permutation problem of frequency-domain blind source separation," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 5, pp. 530–538, 2004.

[15] S. Fagerlund, "Bird species recognition using support vector machines," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 64–64, 2007.

[16] C.-J. Huang, Y.-J. Yang, D.-X. Yang, and Y.-J. Chen, "Frog classification using machine learning techniques," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3737–3743, 2009.

[17] A. Fleury, N. Noury, M. Vacher, H. Glasson, and J.-F. Seri, "Sound and speech detection and classification in a health smart home," in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2008, pp. 4644–4647.

[18] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[19] H. Eghbal-Zadeh, B. Lehner, M. Dorfer, and G. Widmer, "Cp-jku submissions for dcase-2016: A hybrid approach using binaural i-vectors and deep convolutional neural networks," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2016.

[20] M. Valenti, A. Diment, G. Parascandolo, S. Squartini, and T. Virtanen, "Dcase 2016 acoustic scene classification using convolutional neural networks," in *Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2016)*, 2016.

[21] T. Otsuka, K. Ishiguro, H. Sawada, and H. G. Okuno, "Bayesian nonparametrics for microphone array processing," *T-ASLP*, vol. 22, no. 2, pp. 493–504, 2014.

[22] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: a review of recent research," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 356–370, 2012.

[23] J. M. Pardo, X. Anguera, and C. Wooters, "Speaker diarization for multiple distant microphone meetings: mixing acoustic features and inter-channel time differences," *Proceedings of the Ninth International Conference on Spoken Language Processing*, pp. 2194–2197, 2006.

[24] C. Canton-Ferrer, T. Butko, C. Segura, X. Giro, C. Nadeu, J. Hernando, and J. Casas, "Audiovisual event detection towards scene understanding," in *IEEE Computer Society Conference on CVPR Workshops*. IEEE, 2009, pp. 81–88.

[25] M. Cristani, M. Bicego, and V. Murino, "Audio-visual event recognition in surveillance video sequences," *Multimedia, IEEE Transactions on*, vol. 9, no. 2, pp. 257–267, 2007.

[26] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda, "Learning systems of concepts with an infinite relational model," in *AAAI*, vol. 3, 2006, p. 5.

[27] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *Journal of Machine Learning Research*, vol. 9, no. Sep, pp. 1981–2014, 2008.

[28] X. Jin, Y. Zhou, and B. Mobasher, "Web usage mining based on probabilistic latent semantic analysis," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 197–205.

[29] D. V. Pynadath and M. P. Wellman, "Generalized queries on probabilistic context-free grammars," *Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 65–77, 1998.

[30] ——, "Probabilistic state-dependent grammars for plan recognition," in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, 2000, pp. 507–514.

[31] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden markov model: Analysis and applications," *Machine learning*, vol. 32, no. 1, pp. 41–62, 1998.

[32] D. Avrahami-Zilberbrand and G. A. Kaminka, "Incorporating observer biases in keyhole plan recognition (efficiently!)," in *AAAI*, vol. 7, 2007, pp. 944–949.

[33] H. H. Bui, S. Venkatesh, and G. West, "Policy recognition in the abstract hidden markov model," *J. Artif. Int. Res.*, vol. 17, no. 1, pp. 451–499, 2002.

[34] H. H. Bui, "A general model for online probabilistic plan recognition," in *IJCAI*, vol. 3, 2003, pp. 1309–1315.

[35] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 697–706.

[36] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, "Knowledge vault: a web-scale approach to probabilistic knowledge fusion," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 601–610.

[37] C. S. Khoo, S. Chan, and Y. Niu, "Extracting causal knowledge from a medical database using graphical patterns," in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. ACL, 2000, pp. 336–343.

[38] P. Cimiano, A. Hotho, and S. Staab, "Learning concept hierarchies from text corpora using formal concept analysis," *Journal of Artificial Intelligence Research*, vol. 24, pp. 305–339, 2005.

[39] D. Nitti, T. De Laet, and L. De Raedt, "Relational object tracking and learning," in *Robotics and Automation (ICRA), IEEE International Conference*. IEEE, 2014, pp. 935–942.

[40] V.-T. Vu, F. Bremond, and M. Thonnat, "Automatic video interpretation: A novel algorithm for temporal scenario recognition," in *International Joint Conference on Artificial Intelligence*, vol. 3, 2003, pp. 1295–1300.

[41] R. Kojima, O. Sugiyama, and K. Nakadai, "Scene understanding based on sound and text information for a cooking support robot," in *Current Approaches in Applied Artificial Intelligence*. Springer, 2015, pp. 665–674.

[42] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden markov model: Analysis and applications," *Machine Learning*, vol. 32, no. 1, pp. 41–62, 1998.

[43] A. Hashimoto, N. Mori, T. Funatomi, Y. Yamakata, K. Kakusho, and M. Minoh, "Smart kitchen: a user centric cooking support system," in *Proceedings of IPMU*, vol. 8, 2008, pp. 848–854.

[44] E. H. Spriggs, F. De La Torre, and M. Hebert, "Temporal segmentation and activity classification from first-person sensing," in *CVPR Workshops 2009. IEEE Computer Society Conference*, 2009, pp. 17–24.

[45] Y. Yamakata, Y. Tsuchimoto, A. Hashimoto, T. Funatomi, M. Ueda, and M. Minoh, "Cooking ingredient recognition based on the load on a chopping board during cutting," in *IEEE International Symposium on Multimedia*, 2011, pp. 381–386.

[46] Y. Yang, Y. Li, C. Fermuller, and Y. Aloimonos, "Robot learning manipulation action plans by "watching" unconstrained videos from the world wide web," in *The 29th AAAI Conference on Artificial Intelligence*, 2015.

[47] J. Malmaud, J. Huang, V. Rathod, N. Johnston, A. Rabinovich, and K. Murphy, "What's cookin'? interpreting cooking videos using text, speech and vision," in *Proceedings of NAACL'15*, 2015.

[48] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. MIT press Cambridge, 2001, vol. 2.

[49] Y. Inoue and S.-i. Minato, "An efficient method for indexing all topological orders of a directed graph," in *Algorithms and Computation*. Springer, 2014, pp. 103–114.

[50] T. Sato and Y. Kameya, "Parameter learning of logic programs for symbolic-statistical modeling," *Journal of Artificial Intelligence Research*, vol. 15, no. 1, pp. 391–454, 2001.

[51] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying conditional random fields to japanese morphological analysis," in *EMNLP*, vol. 4, 2004, pp. 230–237.

[52] T. Kudo and Y. Matsumoto, "Japanese dependency analysis using cascaded chunking," in *Proceedings of the 6th Conference on Natural Language Learning*, vol. 20, 2002, pp. 1–7.

[53] C. K. Catchpole and P. J. Slater, *Bird song: biological themes and variations*. Cambridge University Press, 2003.

[54] F. Briggs, R. Raich, K. Eftaxias, Z. Lei, and Y. Huang, "The ninth annual mlsp competition: overview," in *Proceedings of the 2013 IEEE International Workshop on Machine Learning for Signal Processing*, 2013, pp. 22–25.

[55] H. Goëau, H. Glotin, W.-P. Vellinga, R. Planqué, and A. Joly, "Lifeclef bird identification task 2016," in *CLEF Working Notes*, 2016.

[56] K. N. Reiji Suzuki, Shiho Matsubayashi and H. G. Okuno, "Localizing bird songs using an open source robot audition system with a microphone array," in *Proceedings of Interspeech 2016*, 2016, pp. 2026–2030.

[57] K. Ryosuke, S. Osamu, i. S. Reij, N. Kazuhiro, and T. Charles, E., "Semi-automatic bird song analysis by spatial-cue-based integration of sound source detection, localization, separation, and identification," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.

[58] P. Aarabi, "The fusion of distributed microphone arrays for sound localization," *Eurasip Journal on Applied Signal Processing*, vol. 2003, no. 4, pp. 338–347, 2003.

[59] J.-M. Valin, F. Michaud, and J. Rouat, "Robust 3d localization and tracking of sound sources using beamforming and particle filtering," in *IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 4. IEEE, 2006, pp. IV–IV.

[60] C. V. Cotton and D. P. Ellis, "Spectral vs. spectro-temporal features for acoustic event detection," in *WASPAA-2011*. IEEE, 2011, pp. 69–72.

[61] Y. Ohishi, D. Mochihashi, T. Matsui, M. Nakano, H. Kameoka, T. Izumitani, and K. Kashino, "Bayesian semi-supervised audio event transcription based on markov indian buffet process," in *ICASSP-2013*. IEEE, 2013, pp. 3163–3167.

[62] M. L. Chin and J. J. Burred, "Audio event detection based on layered symbolic sequence representations," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2012, pp. 1953–1956.

[63] D. Rybach, R. Schlüter, and H. Ney, "Silence is golden: modeling non-speech events in wfst-based dynamic network decoders," in *ICASSP-2012*. IEEE, 2012, pp. 4205–4208.

[64] Y. Sasaki, M. Kaneyoshi, S. Kagami, H. Mizoguchi, and T. Enomoto, "Daily sound recognition using pitch-cluster-maps for mobile robot audition," in *IROS-2009*. IEEE, 2009, pp. 2724–2729.

[65] C. Baugé, M. Lagrange, J. Andén, and S. Mallat, "Representing environmental sounds using the separable scattering transform," in *ICASSP-2013*. IEEE, 2013, pp. 8667–8671.

[66] V. Ramasubramanian, R. Karthik, S. Thiyagarajan, and S. Cherla, "Continuous audio analytics by hmm and viterbi decoding," in *ICASSP-2011*. IEEE, 2011, pp. 2396–2399.

[67] K. Nakamura and K. Nakadai, "Robot audition based acoustic event identification using a bayesian model considering spectral and temporal uncertainties," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2015, pp. 4840–4845.

[68] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: first steps," *Social networks*, vol. 5, no. 2, pp. 109–137, 1983.

[69] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em," *Machine learning*, vol. 39, no. 2-3, pp. 103–134, 2000.

[70] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra, "Clustering on the unit hypersphere using von mises-fisher distributions," *Journal of Machine Learning Research*, vol. 6, no. Sep, pp. 1345–1382, 2005.

[71] S. Sra, "A short note on parameter approximation for von mises-fisher distributions: and a fast implementation of is(x)," *Computational Statistics*, vol. 27, no. 1, pp. 177–190, 2012.

[72] K. Nakadai, H. G. Okuno, H. Nakajima, Y. Hasegawa, and H. Tsujino, "An open source software system for robot audition hark and its evaluation," in *IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2008, pp. 561–566.

[73] G. Schwarz *et al.*, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[74] I. Corona, G. Giacinto, C. Mazzariello, F. Roli, and C. Sansone, "Information fusion for computer security: State of the art and open issues," *Information Fusion*, vol. 10, no. 4, pp. 274–284, 2009.

[75] D. Doran and S. S. Gokhale, "Web robot detection techniques: overview and limitations," *Data Mining and Knowledge Discovery*, vol. 22, no. 1-2, pp. 183–210, 2011.

[76] R. Kosala and H. Blockeel, "Web mining research: a survey," *ACM SIGKDD Explorations Newsletter*, vol. 2, no. 1, pp. 1–15, 2000.

[77] C. Warrender, S. Forrest, and B. Pearlmutter, "Detecting intrusions using system calls: alternative data models," in *Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium*. IEEE, 1999, pp. 133–145.

[78] S.-B. Cho and H.-J. Park, "Efficient anomaly detection by modeling privilege flows using hidden markov model," *Computers & Security*, vol. 22, no. 1, pp. 45–55, 2003.

[79] C. W. Geib and R. P. Goldman, "Plan recognition in intrusion detection systems," in *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01. Proceedings*, vol. 1. IEEE, 2001, pp. 46–55.

[80] N. Lesh, C. Rich, and C. L. Sidner, "Using plan recognition in human-computer collaboration," *Courses and lectures - International Centre for Mechanical Sciences*, pp. 23–32, 1999.

[81] B. A. Goodman and D. J. Litman, "On the interaction between plan recognition and intelligent interfaces," *User Modeling and User-Adapted Interaction*, vol. 2, no. 1-2, pp. 83–115, 1992.

[82] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse, "The lumiere project: Bayesian user modeling for inferring the goals and needs of software users," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1998, pp. 256–265.

[83] F. Jelinek and J. D. Lafferty, "Computation of the probability of initial substring generation by stochastic context-free grammars," *Computational Linguistics*, vol. 17, no. 3, pp. 315–323, 1991.

[84] A. Stolcke, "An efficient probabilistic context-free parsing algorithm that computes prefix probabilities," *Computational linguistics*, vol. 21, no. 2, pp. 165–201, 1995.

[85] T. Sato and Y. Kameya, "Parameter learning of logic programs for symbolic-statistical modeling," *Journal of Artificial Intelligence Research*, vol. 15, pp. 391–454, 2001.

[86] ——, "New advances in logic-based probabilistic modeling by prism," in *Probabilistic Inductive Logic Programming*, 2008, pp. 118–155.

[87] M. F. Arlitt and C. L. Williamson, "Web server workload characterization: The search for invariants," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 24, 1996, pp. 126–137.

[88] "The Internet Traffic Archive," http://ita.ee.lbl.gov/, 2001.

[89] Z. Chi, "Statistical properties of probabilistic context-free grammars," *Computational Linguistics*, vol. 25, no. 1, pp. 131–160, 1999.

[90] T. Sato, "A statistical learning method for logic programs with distribution semantics," in *Proceedings of Internationall Conference on Logic Programming95*, 1995.

[91] L. De Raedt, A. Kimmig, and H. Toivonen, "Problog: a probabilistic prolog and its application in link discovery," in *IJCAI*, vol. 7, 2007, pp. 2462–2467.

[92] T. Sato and P. Meyer, "Infinite probability computation by cyclic explanation graphs," *Theory and Practice of Logic Programming*, pp. 1–29, 2013.

[93] C. S. Wetherell, "Probabilistic languages: a review and some open questions," *ACM Computing Surveys (CSUR)*, vol. 12, no. 4, pp. 361–379, 1980.

[94] L. Huang, "Advanced dynamic programming in semiring and hypergraph frameworks," *COLING*, 2008.

[95] M.-J. Nederhof and G. Satta, "Computation of infix probabilities for probabilistic context-free grammars," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1213–1221.