BEYOND LEXICAL MEANING: PROBABILISTIC MODELS FOR SIGN LANGUAGE RECOGNITION

SYLVIE C.W. ONG

(B Sc. (Hons) (Electrical Engineering), Queen's University, Canada)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2007

Acknowledgements

I am truly indebted to my supervisor, Assoc. Prof. Surendra Ranganath for his continuous guidance and support during this work, and for always encouraging me to achieve more.

I am deeply indebted to the National University of Singapore for the award of a research scholarship.

I would also like to thank members of the Deaf & Hard-of-Hearing Federation (Singapore), for providing signing data and for showing me the finer points of sign language, especially Judy L.Y. Ho, Edwin Tan, Lim Lee Ching, Hairaini Ali, Tom Ng, Wilson Ong, and Ong Kian Ann.

On a personal note, I would like to thank my parents for their endless love and support and unwavering belief in me. My extreme gratitude also goes to my friends and neighbours who fed and sheltered me in my hour of need.

Sylvie C.W. Ong

15 April 2007

Contents

Summary

This thesis presents a probabilistic framework for recognizing multiple simultaneously expressed concepts in sign language gestures. These gestures communicate not just the lexical meaning but also grammatical information, i.e. inflections that are expressed through systematic spatial and temporal variations in sign appearance. In this thesis we present a new approach to analyse these inflections by modelling the systematic variations as parallel information streams with independent feature sets. Previous work has managed the parallel complexity in signs by decomposing the sign input data into parallel data streams of handshape, location, orientation, and movement. We extend and further generalize the concept of parallel and simultaneous data streams by also modelling systematic sign variations as parallel information streams. We learn from data, the probabilistic relationship between lexical meaning and inflections, and the information streams; and then use the trained model to infer the sign meaning conveyed through observing features in multiple data streams.

We show how to take advantage of commonalities between how grammatical processes affect appearances of different root sign words to reduce parameters learned in the model and recognize new and unseen combinations of root words and grammatical information. This is crucial because there is a large variety of information that can be conveyed in addition to the lexical meaning in signs and hence a large variety of appearance changes that can occur to a root word. It is therefore crucial to be able to recognize unseen new signs conveying new combinations of lexical and grammatical information.

In preliminary experiments, we recognize isolated gestures using a Bayesian network (BN) to combine the information stream outputs and infer both the basic lexical meaning and the inflection categories. In further experiments, we apply our approach to recognize continuously signed sentences containing inflected signs. Continuous signing presents additional challenges as the segmentation of a continuous stream of signs into individual signs is a difficult problem. We propose a novel dynamic Bayesian network (DBN) structure – the Multichannel Hierarchical Hidden Markov Model (MH-HMM) for continuous sign recognition. Just as in the case for the BN, the MH-HMM models the probabilistic relationship between lexical meaning and inflections, and the information streams. Sentences are implicitly segmented into individual signs during the recognition process, while synchronization between multiple streams is obtained through the novel use of a synchronization variable in the network structure. The vocabulary used in the continuous signing experiments is very complex. The vocabulary size is 98 signs, with 73 different sentences appearing in the training and test set data. The 98 signs are made up of combinations of 29 lexical meanings, and two different types of inflections, one with 11 distinct values and the other with 3 distinct values. Many of the root sign words appear in multiple variations due to inflections. For example, the root sign word GIVE appears in 16 different versions. Some of the inflections modify the sign simultaneously, further increasing the complexity of the vocabulary.

Computational complexity of inferencing in DBNs increases with network size. We show how to use particle filtering as an approximate inferencing algorithm to manage the computational complexity for our proposed DBN model. Experimental results demonstrate the feasibility of using the MH-HMM for recognizing inflected signs in continuous sentences. We also demonstrate results for recognizing continuously signed sentences containing unseen new signs.

List of Tables

List of Figures

Chapter 1

Introduction and background

Sign language (SL) communication is a richly expressive medium that involves not only hand/arm gestures (for manual signing) but also non-manual signals (NMS) conveyed through facial expressions, head movements, body postures and torso movements. NMS is most used for syntactic constructions, for example, to mark topics, relative clauses, negative clauses, and questions [94]. In manual signing, the interplay of grammatical elements and lexical meaning produces a large number of complex variations in sign appearances [94]. In SL, many of the grammatical processes involve systematically changing the manual sign appearance to convey information in addition to the lexical meaning of the sign. This includes information that would usually be expressed in English through prefixes and suffixes or additional words like adverbs. Hence, while information is expressed in English by using additional words as necessary rather than changing a given word's form, in SL, it is often expressed through a change in the form of the root sign word. Thus, just as there is a large variety of prefixes, suffixes, and adverbs that may be used with a particular word in English, there is also a large variety of different systematic appearance changes that can be made to a root word in SL.

In this thesis we are concerned with SL recognition. The term **SL recognition** refers to extracting information from the signed data stream (for example of a sentence), and recognizing the sequence of manual signs and NMS in that stream. The output of the recognition process is the sequence of meanings (words and grammatical information) conveyed in the signing sequence. This is a very raw form which is not grammatical, and may not have a one-to-one mapping with the words of any spoken language. Thus, a complete sign-to-text/speech translation system would additionally require machine translation from the recognized sequence of meanings to the text or speech of a spoken language such as English. Machine translation is usually not addressed in SL recognition work, and is beyond the scope of this thesis.

Much of SL recognition research has focused on solving problems similar to those that occur in speech recognition, such as scalability to large vocabulary, robustness to noise and person independence, to name a few. These are worthy problems to consider and solving them is crucial to building a practical SL recognition system. However, the almost exclusive focus on these problems has resulted in systems that can only recognize the lexical meanings conveyed in signs, and bypass the richness and complexity of expression inherent in manual signing.

This thesis is a step towards addressing the imbalance in focus. In taking this first step, it is necessary to limit the scope to manual signing. So although NMS is an important part of SL communication, NMS and its recognition is not considered in any detail. The focus of this work is on recognizing the different sign appearances formed by modulating a root word and extracting both the lexical meaning and the additional grammatical information that is conveyed by the different appearances.

Specifically, the focus is on modelling and extracting information conveyed by two types of grammatical processes that produce systematic changes in manual sign appearance, viz., **directional use of verbs** and **temporal aspect inflections**. These processes will be described in more detail in the next section (Section 1.1). The signs and grammar described are with reference to American Sign Language (ASL) because it is one of the most well-researched sign languages – by sign linguists as well as by researchers in machine recognition. Its grammatical rules have been studied extensively and well-documented in comparison with many other sign languages in use around the world. One of the motivations for SL recognition research is the contributions that it can make to gesture recognition research in general. In Section 1.2, the connection between speech-accompanying gesticulations and SL manual signing is considered, especially as it pertains to the grammatical processes mentioned above. Section 1.3 describes more fully the motivation of our research, followed by a statement of the research goals in Section 1.4.

For the rest of this thesis, unless otherwise noted, the terms **word** and **sign** shall refer exclusively to manual signing and do not include NMS. Our definitions of these two terms are given below. They do not necessarily reflect accepted conventions in SL linguistic literature and thus should be considered as only applicable within the scope of this thesis. If the lexical/word meaning *and* grammatical information conveyed by two SL hand gestures is the same, then we consider it to be the same **sign**. However, gestures that convey the same lexical/word meaning but different grammatical information are defined to be the same **word** but different and distinct signs. So for example, the same word inflected in different ways results in different signs.

1.1 Sign language communication

As mentioned above, most research work in SL recognition has focused on classifying the lexical meaning in signs. This is understandable since the lexical information in signs does express the main information conveyed through signing. For example, by observing the hands in the sequence of Figure 1.1, we can decipher the lexical meaning conveyed as 'YOU STUDY'[1]. However, without observing NMS and the repetitiveness of the movement in the signing, we cannot decipher the full meaning of the sentence as, "Are you studying very hard?". The query in the

[1] Words in capital letters are sign glosses which represent signs with their closest meaning in English. However, the signs do not necessarily correspond exactly in meaning with the glosses that represent them.

sentence is expressed by the body leaning forward, head thrust forward and raised eyebrows towards the end of the signed sequence (e.g. in Figure 1.1(e),(f)). To refer to an activity performed with great intensity, the lips are spread wide with the teeth visible and clenched; this co-occurs with the sign STUDY. In addition to information conveyed through these NMS, the sign is performed repetitively, tracing a circular path in 3-dimensional space, with smooth motion. This continuous action further distinguishes the meaning as “studying” instead of “study”. In the following sections, issues related to the lexical form of signs will be considered first, followed by some pertinent issues with respect to modifications of signs that carry grammatical meaning.

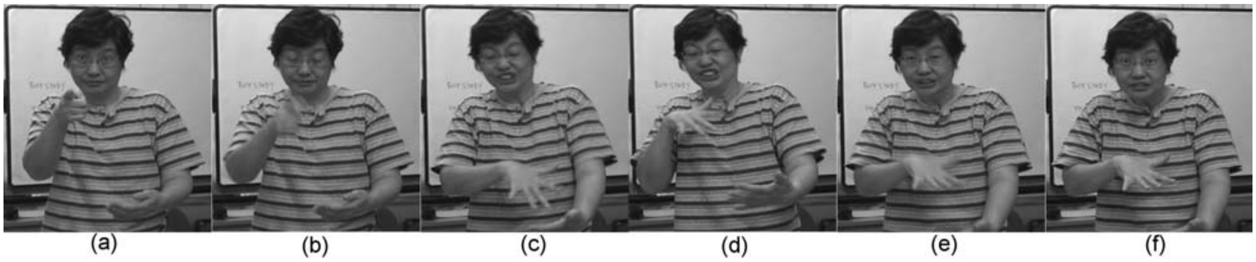


Figure 1.1: A sequence of video stills from the sentence translated into English as “Are you studying very hard?”. Frame (a) is from the sign YOU. Frames (c)–(f) are from the sign which contains the lexical meaning STUDY. Frame (b) is during the transition from YOU to STUDY.

1.1.1 Manual signs to express lexical meaning

Sign linguists agree that signs have internal structure that can be broken down into smaller parts [152], and they generally distinguish the basic parts as consisting of the handshape, hand orientation, location and movement. Handshape refers to the finger configuration, orientation to the direction in which the palm and fingers are pointing, and location to where the hand is placed relative to the body. Hand movement includes both path movement that traces out a trajectory in space, and movement of the fingers and wrist. Each of these parts have a limited number of possible categories, or "primes" (for example [14] identifies 40 distinct handshapes, 16-18 distinct orientations, 12 distinct locations, and 12 simple movements).

Two major ways of analysing the sign structure are: 1) as temporally parallel phenomena where signs are primarily seen as a simultaneous organization of features; or 2) as primarily sequential phenomena where signs are organized as a sequence of temporal segments [95]. In Stokoe's [144] representation, a sign is described as a combination of simultaneous values for location, oriented handshape, and one or more movements. If there are sequences of handshapes, locations, and orientations within a sign, these are considered as by-products of the movement component. In Liddell's representation [94], [95], signs consist of movement and hold segments that are produced sequentially. Movement segments are defined as periods during which some part of the sign is in transition, whether handshape, location or orientation. Hold segments are periods when all these parts are static. Movement segments have additional features, including path contour or path shape (the shape of the path traced in 3-dimensional space by the hand); contour plane (the 2-dimensional plane in which the path is traced in); and other movement path attributes like shortening, acceleration, reduction or enlargement. Many of the recent models also propose sequential representation of signs ([27],[125],[137],[164]).

An important phenonemon that occurs in continuous signing is movement epenthesis. When signs occur in a continuous sequence to form sentences, the hand(s) need to move from the ending location of one sign to the starting location of the next. Simultaneously, the handshape and hand orientation also change from the ending handshape and orientation of one sign to the starting handshape and orientation of the next. These inter-sign transition periods are called movement epenthesis [95] and are not part of either of the signs. Figure 1.1(b) shows a frame within the movement epenthesis where the right hand is transiting from performing the first sign to the second sign in the sentence. In continuous signing, processes with effects similar to co-articulation in speech also do occur, where the appearance of a sign is affected by the preceding and succeeding signs (e.g. hold deletion, metathesis and assimilation [152]). However, these processes do not necessarily occur in all signs; for example, hold deletion is variably applied depending on whether the hold involves contact with a body part [95]. Hence movement epenthesis occurs most frequently during continuous signing and should probably be tackled first by machine analysis, before dealing with the other phonological processes.

The systematic changes to the sign appearance during continuous signing described above (addition of movement epenthesis, hold deletion, metathesis, assimilation) do not change or add to the sign meaning. However, there are other systematic changes to one or more parts of signs which affect the sign meaning. Two of these types of modulatory processes are briefly described in the next two sections.

1.1.2 Directional verbs

Directional verbs are made with various handshapes and movement path shapes to encode the lexical meaning of the verb. Meanwhile, the movement path direction (the direction in which the hand is moving in 3-dimensional space) serves as a pointing action to identify the subject and the object of the verb [94].

Example 1. Figure 1.2 (a) shows the appearance of the sign which has lexical meaning TEACH and with subject and object being the signer and the addressee, respectively (English translation: “I teach you”). Figure 1.2 (b) shows the sign with the same lexical meaning of TEACH, this time with subject and object being the addressee and the signer, respectively (English translation: “You teach me”). In Figure 1.2 (c), the subject of the verb is indicated as the signer. The object is neither the signer nor the addressee but a third person who could either be someone standing (off-camera) roughly to the left of the signer, or a non-present person. In the second case, the signer would have already set up or established

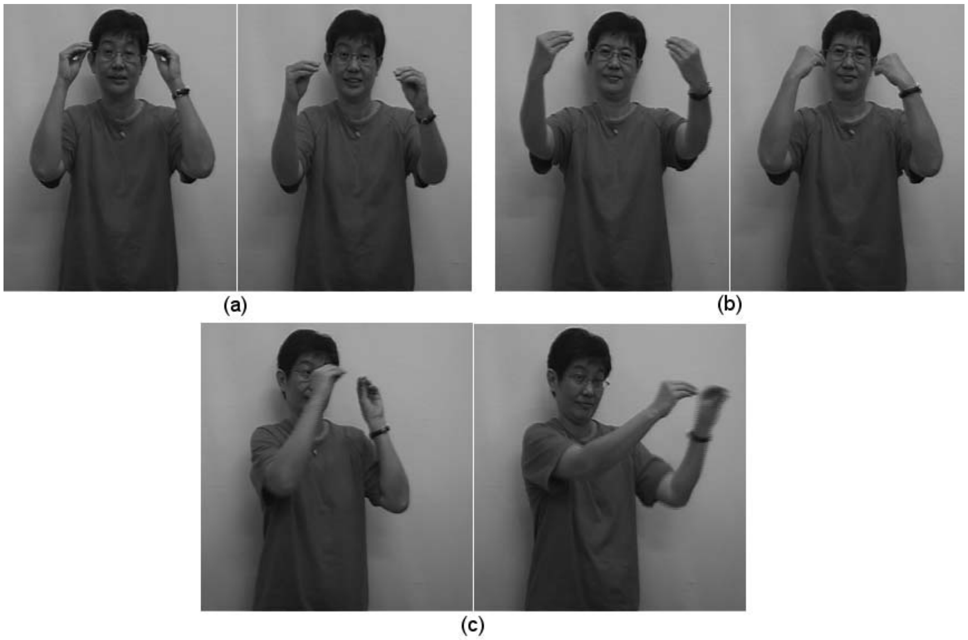


Figure 1.2: The sign TEACH pointing towards different subjects and objects : (a) “I teach you”, (b) “You teach me”, (c) “I teach her/him (someone standing to the left of the signer)”.

this non-present referent in the location to the left of her body. One of the ways of doing this is by using a pronoun to point to that location right after making the sign for the referent (e.g. the person’s name) [8]. (We will use this method of establishing referents in the experiments of Chapter 6). Once established, pointing signs can be made in the direction of the location just as if the referent really was present there.

The modulations in movement path direction as described above are examples of directional verb inflections. There are a few things to note about directional verbs. The addressee or any other referent could be located just about anywhere with respect to the signer. Thus the directionality of these verbs is not fixed, but varies depending on the actual location of the entity it is directed towards or the established referent location (in subsequent analysis we shall only refer to the case where the referent is physically present, with the understanding that the analysis would apply equally to the case of the non-present referent). The hand can point in an unlimited number of directions, and Liddell [94] makes a convincing argument that this directional use of signs does not convey symbolic information but instead conveys the same information as pointing co-verbal gestures. In spoken language the phonetic signal that conveys symbolic information (i.e. the lexical word meaning) is expressed verbally, while pointing co-verbal gestures would be performed by the hand/arm, which are completely separate and distinct articulators than that for speech. In the case of SL discourse, the symbolization and the pointing both occur through movements of the hands and body. It is important however to distinguish the two functions as separate within the same sign.

Another key fact to note is that movement direction modulation is accompanied by location change and often also a change in palm orientation. Although the final location of the hand, for example, is not describable in terms of a fixed set of phonological or phonetic features, it does depend on the locations of entities these verbs are directed towards and the signer's judgement in tracing a path that leads from the starting point of the sign towards the entity that is the verb's object. We will make use of this fact for modelling and in experiments described in Chapter 4 and 6, respectively.

Lastly, the direction of the signer's eye gaze (and frequently his/her head position) is also important for understanding the grammatical role of different referents in the sentence [8]. This NMS is however beyond the scope of the thesis and will not be addressed here.

1.1.3 Temporal aspect inflections

In the sentence of Figure 1.1, the sign STUDY expresses aspectual information in addition to the lexical meaning of the verb. The handshape of this inflected sign is the same as in its uninflected form but the movement of the sign is modified to show how the action (STUDY) is performed with reference to time. The English translation for this sign would be "studying continuously" or "studying for a while". This particular inflection value is denoted as [DURATIONAL]. Examples of other signs that can be inflected in this way are WRITE, SIT, LOOK-AT and 33 other signs listed by Klima and Bellugi in [81]. Below are some examples and illustrations of the [DURATIONAL] inflection as well as other inflections in the same category, collectively called temporal aspect inflections.

Example 2. In Figure 1.3(a), the sign is uninflected and conveys the lexical meaning LOOK-AT. It has a linear, straight movement path shape. In Figure 1.3(b), the sign is modulated with the [DURATIONAL] inflection to give the meaning "look at continuously". Similar to the inflected sign for STUDY mentioned above, here the sign is also performed repetitively in a circular path shape

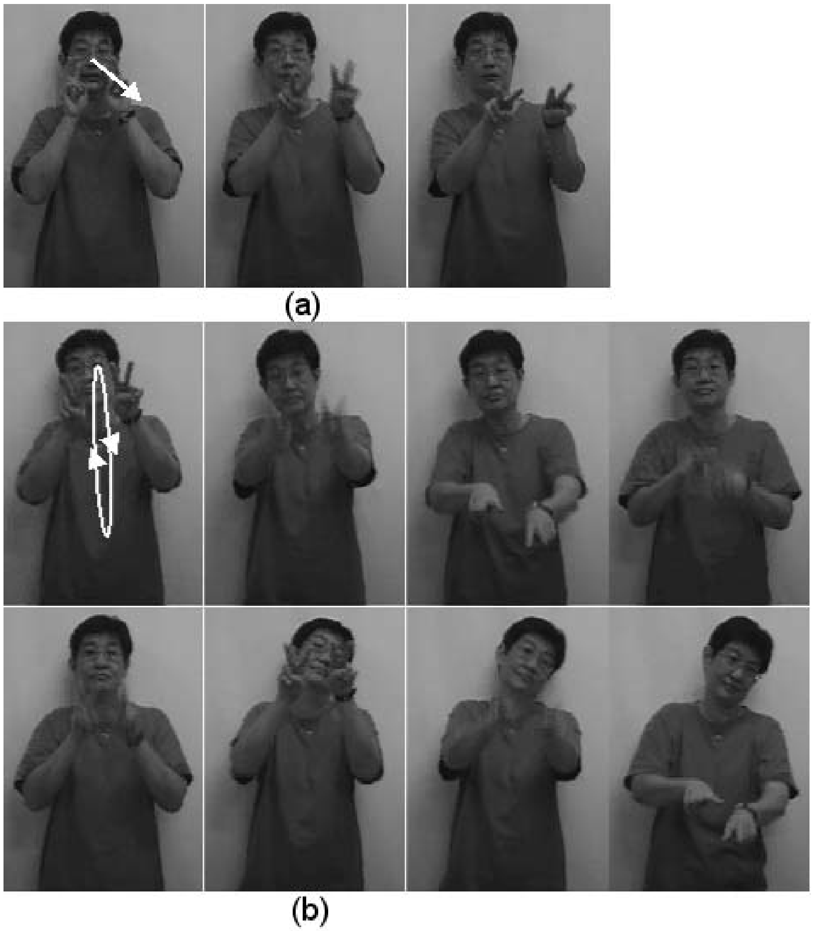


Figure 1.3: (a) The sign LOOK-AT (without any additional grammatical information), (b) the sign LOOK – AT$^{[\text{DURATIONAL}]}$, conveying the concept “look at continuously”.

with smooth motion.

Example 3. In Figure 1.4(a), the sign is uninflected and conveys the lexical meaning CLEAN. In Figure 1.4(b), the sign is modulated with the [INTENSIVE] inflection to give the meaning “very clean”. Compared to the unmodulated sign, the movement in CLEAN$^{[\text{INTENSIVE}]}$ is faster and bigger, and the hand/arm is more tense. FAST and AFRAID are examples of other signs that can be modulated in

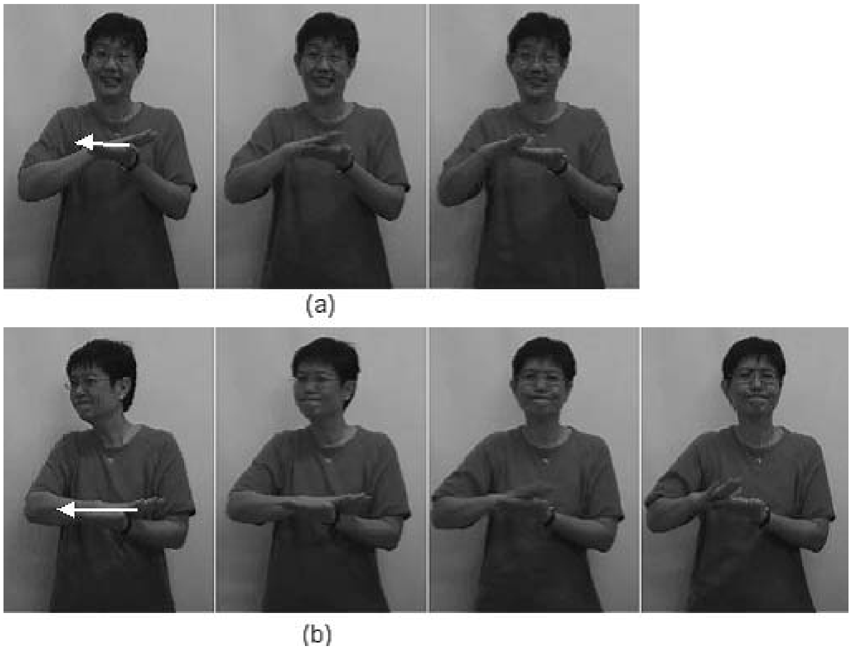


Figure 1.4: (a) The sign CLEAN (without any additional grammatical information), (b) the sign CLEAN$^{[INTENSIVE]}$, conveying the concept “very clean”.

this way.

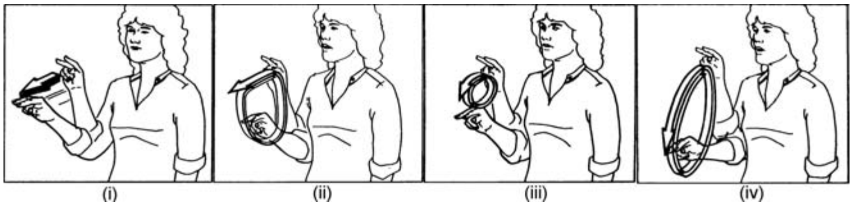


Figure 1.5: Signs with the same lexical meaning, ASK, but with different temporal aspect inflections (from [126]) (i) [HABITUAL], meaning “ask regularly”, (ii) [ITERATIVE], meaning “ask over and over again”, (iii) [DURATIONAL], meaning “ask continuously”, (iv) [CONTINUATIVE], meaning “ask for a long time”.

Figure 1.5 shows illustrations of the signs expressing the lexical meaning ASK, with different types of aspectual inflections - [HABITUAL], [ITERATIVE], [DURATIONAL], and [CONTINUATIVE].

From these examples we can see that these modulations firstly affect the movement path shape and size (both of which also affect the hand location, a fact that we use to advantage in sign modelling and in experiments of Chapter 4 and 6, respectively), and secondly, the movement rhythm and speed. An example of modulations of the latter type is CLEAN$^{\text{[INTENSIVE]}}$ which has a faster movement than the uninflected word sign CLEAN. The [DURATIONAL] and [HABITUAL] inflections induce smooth motion at a constant rate while the [CONTINUATIVE] and [ITERATIVE] inflections induce uneven motion (unfortunately these differences in rhythm and speed are difficult to illustrate on the printed page). Sign linguists postulate that all the variations due to expression of aspectual meanings differ from one another in only a limited number of spatial and temporal dimensions, each with a small number of contrastive values [81]. These dimensions are: *rate* (relatively fast or slow), onset-offset *hold* (the movement can start or end with a hold), *tension* (presence or absence of tension in the hand/arm), *evenness* (constant or uneven rhythm), *size* (relatively large or small), *contouring* (straight, circular, elliptical) and number of *cycles* (single or multiple).

The meanings conveyed through these modulations in movement are associated with aspects of the verbs that involve frequency, duration, recurrence, permanence, and intensity [81],[126]. Besides the examples mentioned above, other meanings that may be conveyed include “incessantly”, “from time to time”, “starting to”, “increasingly”, “gradually”, “resulting in”, “with ease”, “readily”, “approximately” and “excessively”. Klima and Bellugi [81] lists 11 different types of aspectual meanings that can be expressed. The important thing to note is that the aspectual information is conveyed in addition to and without changing the lexical meaning of the verb or adjective.

Lastly, signs marked for aspectual meaning tend to appear with specific non-manual signals, including specific facial expressions as well as head positions and movements [94]. However NMS is not addressed here.

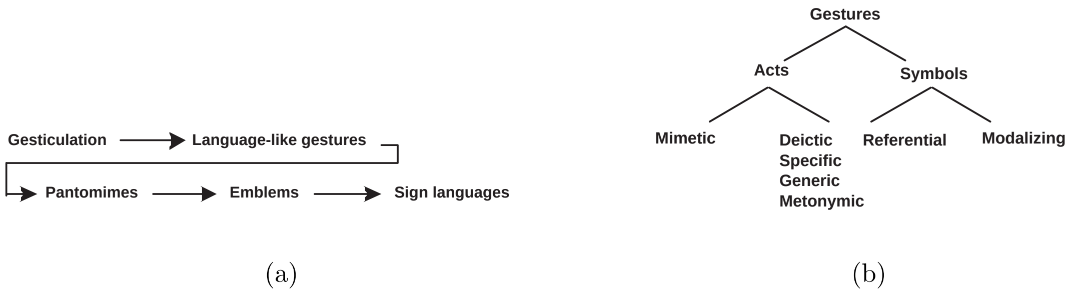
1.1.4 Multiple simultaneous grammatical information

In ASL, multiple grammatical information may be conveyed through a single sign, by creating complex spatio-temporal sign forms [81]. The modulations of sign movement due to different categories of grammatical processes affect different characteristics of movement. For example, a directional verb points to its subject and object through the direction of the movement. Whereas, if the verb is marked for aspectual meaning, this is expressed through the movement path shape, size and speed. Each of these characteristics is mutually exclusive and their “values” can combine in parallel. So for example, we can express the meaning “you give to me regularly” as distinct from “you give to me continuously” or “I give to you regularly” and so on. Each modulation category adds grammatical information to the sign. The appearance of a sign can reflect the effects of several coexisting interrelated systems [81]: 1) a lexical system, 2) a pointing system, and 3) the aspectual inflectional system. Each of these systems utilizes certain selected properties of space, form, and movement that are unique to, or especially characteristic of that system.

In the modelling and experiments on isolated gestures in Chapter 3, and on continuous signing in Chapters 4 and 6, signs that carry multiple simultaneous grammatical information will be considered.

1.2 Gestures and sign language

In taxonomies of communicative hand/arm gestures, SL is often regarded as being the most structured, with the most symbolic content and rigidly defined conventions among all the gesture categories. In the continuum of gestures described by Kendon, sign languages are at the opposite end of the scale from gesticulation (Figure 1.6(a) [77], [104]). A main distinction made in gestures is whether it is an autonomous gesture or a gesticulation. Autonomous gestures are performed in the absence of other modes of communication (usually speech). They are standardized, symbolic gestures that are complete within themselves [77], [163]. In contrast, gesticulations are typically not performed on their own, but along with speech. The verbal part conveys lexical and grammatical information, while the accompanying gesticulation depicts non-symbolic information, for example actions or spatial relationships [76], [129]. In such a dichotomy, sign languages would be firmly placed in the category of autonomous gestures, the argument being that in the absence of speech (and forgetting NMS for the moment) manual signing necessarily carries all the lexical and grammatical information conveyed in the language [128]. Manual signs are complete within themselves, and no other concurrent mode of communication is required. However, this does not mean that *all* the information conveyed in manual signing is lexical and grammatical information. Manual signing does indeed include symbolic content but this content is not *all* that it includes. Signs can also convey the same information as in speech-accompanying gesticulations; some elements in SL signs serve the same function and/or have the same form as gesticulations.



(a) (b)

Figure 1.6: Two different gesture taxonomies ([128]): (a) Kendon's continuum [104], (b) Quek's taxonomy [128].

Kendon [78] describes the main role of gesticulations as being spatial/temporal qualifiers that specify location, orientation, spatial relation and shape, or as a volumetric qualifier that specifies size. Quek [128] distinguishes between acts, which are gestures whose movements relate directly to the intended interpretation (iconic, pantomimic or deictic), and symbols, which are gestures whose forms are arbitrary in nature (refer to Figure 1.6(b)). Acts can be of four classes [129]:

- **Locative gestures** point to a location or to an object.
- **Orientational gestures** show placement of objects by specifying rotations of the hand.
- **Spatial pantomimes** use the hand movement trajectory to depict some shape, path or spatial outline.
- **Relative spatial gestures** show spatial relationships such as nearer, further, further right, etc.

To this list perhaps we can add one more class – **temporal pantomimes** – gestures that use the movement dynamics (speed and acceleration) of the hand to depict the duration, frequency, manner, and repetitiveness (collectively called the *temporal contour*) of an action.

There are a few types of signs which exhibit the form, function or both, of the gesticulations and act gestures described above. Some of these are described below with reference to ASL signs and grammar.

1.2.1 Pronouns and directional verbs

Pronouns are made with the handshape of the extended index finger, and a straight-line movement path shape. Directional verbs are made with various handshapes and movement path shapes to encode the lexical meaning of the verb (see Section 1.1.2). What both these types of signs have in common is that they point, either at objects or in the direction of some location which has been established as representing a referent [94]. The pointing action identifies the person referred to in the case of pronouns. In the case of directional verbs, the pointing action identifies the subject and the object of the verb. Thus there are both symbolic and *deictic* elements in these signs, and they fulfill functions associated with the **locative gesture** class mentioned above.

1.2.2 Temporal aspect inflections

These inflections modulate spatial and dynamic (speed and acceleration) characteristics of sign movements to express a temporal contour (i.e. the duration, frequency, manner and repetitiveness) in a verb or adjective (see Section 1.1.3). Klima and Bellugi [81] have proposed that the temporal contour of the action is reflected in the spatial and dynamic characteristics of the signs' movement: "The modulatory forms are not incongruent and are in some sense indicative with their meanings: permanent or enduring states are characterized by continuous movements, recurring states by repeated end-marked movements, intensification of a state or quality by tense rapid movement." These modulated signs would therefore seem to have *iconic* elements (the sign form suggests its meaning) and we could perhaps call this the **temporal pantomimes** gesture class.

1.2.3 Classifiers

These signs can function in many different ways [8] including, illustrating the precise and relative locations, orientations and/or actions of two referents, by positioning the hands in particular locations in space and moving them in relation to each other; moving the hands to mimic the actions of the objects that they represent; indicating the shape and size of an object by tracing its outline with the hands. Classifiers would seem therefore to fulfill many of the functions described in the classes of act gestures above, including that of **orientational gestures**, **spatial pantomimes** and **relative spatial gestures**.

The fact that SL signs are autonomous gestures does not mean that they cannot incorporate forms and functions of gesticulations. As the above descriptions illustrate, signs can have both functions attributed to autonomous gestures (symbolic) as well as that attributed to gesticulations (act). In the case of the directional use of verbs and temporal aspect inflections – the two categories of grammatical processes that are the focus of the modelling and recognition framework presented in this thesis – the information conveyed is quite different from that conveyed through gesticulations. In directional verbs, the subject and object of verbs are identified and in signs marked for temporal aspect, the temporal contour of actions is conveyed. Whereas, the information conveyed through gesticulations usually pertains to the specification of location, orientation, spatial relation, shape, and size. However, we argue that the form in which the modulations due to grammatical processes expresses itself shares some of the same dimensions or features sets as gesticulations. Directional verbs point and are *deictic*, just like the location gestures mentioned in Quek's taxonomy ([129]). Signs marked for temporal aspect exhibit spatial and temporal variation in movement path and dynamics (speed and acceleration) that are not mentioned in Quek's analysis. However, it has *iconic* elements and we can imagine how a speech-accompanying gesticulation might be made quickly in a tense manner in order to convey a sense of urgency or emphasis. The key issue is that although the information conveyed is not the same, the pointing action and movement dynamics that are expressed are similar, and in both SL and gesticulations, the pointing action and movement dynamics are conveying information.

1.3 Motivation of the research

There are two main motivations for SL recognition research. Firstly, there are many useful and practical applications that can be made possible as a result, and secondly because of the contributions it can make to gesture recognition research in general.

One of the applications envisioned for SL recognition is of course in a signing-to-text/speech translation system. In an ideal system, the SL recognition module would have a large and general vocabulary, be able to capture and recognize manual sign information and NMS, perform accurately in real-time and robustly in arbitrary environments, and allow for maximum user mobility. Such a translation system is not the only use for SL recognition systems however, and other useful applications where the system requirements and constraints may be quite different, include the following:

- Translation or complete dialog systems for use in *specific transactional domains* such as government offices, post offices, cafeterias, etc. [5],[103],[135],[97]. These systems may also serve as a *user interface to PCs* or information servers [11]. Such systems could be useful even with limited vocabulary and formulaic phrases, and a constrained data input environment (perhaps using direct-measure device gloves [46],[135] or colored gloves and constrained background for visual input [5]).

- Bandwidth-conserving *communication between signers through the use of avatars*. Sign input data recognized at one end can be translated to a notational system (like HamNoSys) for transmission and synthesized into animation at the other end of the channel. This represents a great saving in bandwidth as compared to transmitting live video of a human signer. This concept is similar to a system for computer-generated signing developed under the Visicast project ([79]) where text content is translated to SiGML (Signing Gesture Markup Language, based on HamNoSys) to generate parameters for sign synthesis. Another possibility is creating SL documents for storage of recognized sign data in the form of sign notations, to be played back later through animation.

- Automated or semi-automated *annotation of video databases of native signing.* Linguistic analyses of signed languages and gesticulations that accompany speech require large-scale linguistically annotated corpora. Manual transcription of such video data is time-consuming, and machine vision assisted annotation would greatly improve efficiency. Head tracking and handshape recognition algorithms [116], and sign word boundary detection algorithms [83] have been applied for this purpose.

One of the most difficult goals in gesture recognition research is the recognition of 'natural' gestures or gesticulations - spontaneous, free-form gestures that often accompany verbal discourse (see Section 1.2) [76], [40]. Natural gestures are distinct from the synthetic gestures in use by many human-computer interaction applications. The latter usually use a small vocabulary of artificially defined gestures that are designed to be easily and reliably recognized [40]. Natural gestures, being free-form, are infinitely variable and thus much more challenging to recognize than synthetic gestures. As noted in Section 1.2, many manual signs in SL exhibit the same form, function or both, as these natural gestures. Pronouns, directional verbs, signs marked for temporal aspect, and classifiers contain *non-symbolic*, *iconic*, *deictic* and *pantomimic* elements – these signs have characteristics that relate directly to the intended interpretation. Furthermore, signs obviously share the same articulators as natural gestures – the hands and arms. So signs and natural gestures exist in the same visual medium, and can perform similar functions and convey similar information. The key difference however is that SL signs are much more structured and SL recognition has a clear, measurable goal, that of recognizing the word meaning and grammatical information conveyed by the signer. This makes SL recognition a good starting point for developing methods to recognize natural gestures. SL signs can be a good test-bed and useful benchmark for evaluating gesture recognition systems and proposed frameworks. It has a naturally developed complexity and a large well-defined vocabulary for obtaining data with a known ground truth. Achieving the goal of automatic machine recognition of this data requires addressing all the complexities inherent in SL.

In the Gesture Workshop of 1997, Edwards identified two aspects of SL recognition that had often been overlooked by researchers – facial expression, and the use of space and spatial relationships in signing [40]. Since then, although there has been some work to tackle these aspects, the focus of research continues to be elsewhere and hence progress has been limited. SL recognition research to-date has mostly produced systems that only recognize the lexical meanings conveyed in signs, missing out on important information conveyed through deictic and iconic characteristics in signs. For a practical and useful application that is based on SL recognition, this is unacceptable. Another reason for shifting the focus to characteristics in manual signing that are not purely symbolic is that this is precisely where SL signs and natural gestures intersect in form and function and focusing on recognizing these characteristics would represent concrete steps towards natural gesture recognition. By addressing the modelling and extraction of information from directional verbs, and signs marked for temporal aspect – signs that have deictic and iconic characteristics – we hope that the work in this thesis would represent just such steps.

1.4 Goals

The goal of this thesis is to recognize signs which convey information in addition to lexical/word meaning. This information is conveyed through grammatical processes that produce systematic changes in sign appearance. We seek to first model how the lexical and grammatical information conveyed affect the sign appearance, then use this model to extract that information from observations of signing data. The focus will be specifically on modelling directional use of verbs and temporal aspect inflections (see Section 1.1.2 and 1.1.3). These two categories of grammatical processes may in fact appear in parallel, simultaneously affecting the sign appearance (see Section 1.1.4). Thus such simultaneous modulations on signs should also be modelled and the simultaneously conveyed information should likewise be extracted.

The goal of much previous research in SL recognition is scalability to large vocabulary – i.e. being able to recognize a large number of lexical words. In contrast, one of the requirements of our proposed model is to be able to recognize a large number of *combinations* of lexical words and grammatical information. This is crucial because there is a large variety of information that can be conveyed in addition to the lexical meaning in signs and hence a large variety of appearance changes that can occur to a root word. It is not possible to obtain training data for *all* these appearances, hence ideally the model should be able to recognize unseen signs conveying new combinations of lexical and grammatical information.

The sentences used in the experiments on recognizing continuous signing in Chapter 6 were obtained from a signer who is a deaf individual and a native signer of the local (Singaporean) sign language. We felt that it was important to work closely with the local deaf community and to elicit their input and help in obtaining experimental data. At present many recognition results reported in the literature do not use data from native signers or even deaf individuals. Some exceptions are Imagawa et al. [67], Vogler [157], and Tamura and Kawasaki [148], while Tanibata et al. [149] used a professional interpreter. As mentioned by Braffort [25], the goal of recognizing signing as it is used in communication among deaf individuals requires close collaboration with native signers and SL linguists.

1.5 Organization of thesis

The rest of this thesis is organized as follows. Chapter 2 presents a literature review, as well as our model for signs that convey grammatical information and an overview of our proposed approach. Chapter 3 presents the framework and experimental results in recognizing a simulated vocabulary of isolated gestures with Bayesian networks. This is extended to recognition of continuous signing with dynamic Bayesian networks, and this framework is presented in Chapter 4. Inferencing in dynamic Bayesian networks (DBN) is the subject of Chapter 5 with particular attention to approximate inferencing with sampling methods as a way of dealing with the computational complexity in the DBN models for continuous sign recognition. Experimental results using these inference techniques are presented in Chapter 6. Chapter 7 concludes the thesis by presenting the research contributions and directions for future work.

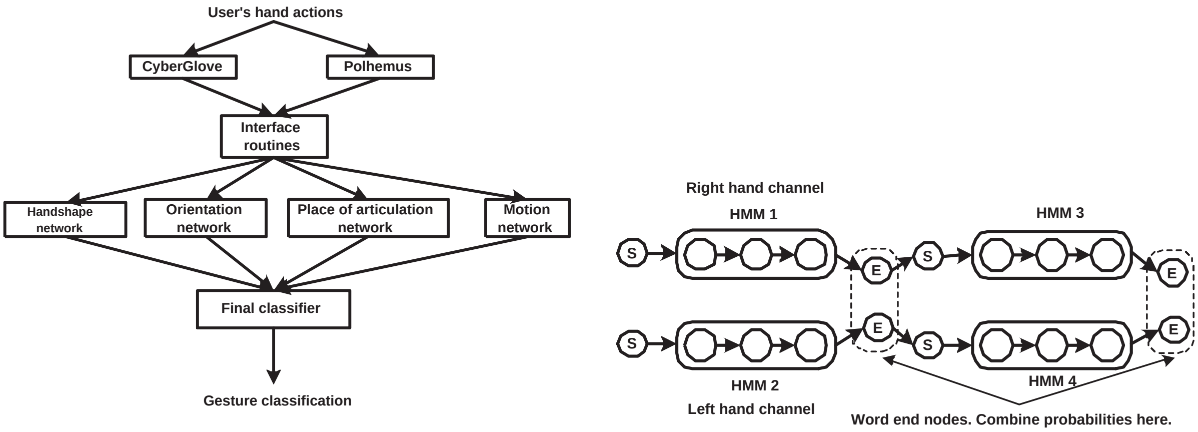
Chapter 2

Review and overview of proposed approach

2.1 Related work

The two main approaches to manual sign classification either employ a single classification stage to classify the whole sign, or represent the sign as consisting of simultaneous components, classify the components individually and then integrate them together for sign-level classification. Figure 2.1 shows examples of the latter approach. Figure 2.1(a) ([153]) is a block diagram of the two-stage classification scheme while Figure 2.1(b) ([157]) shows sign components modelled as separate hidden Markov model (HMM) channels. Various classification methods have been used to either classify the sign directly or classify one of the sign components. These methods include neural networks (NN) and variants [6, 41, 47, 57, 66, 80, 107, 145, 154, 159, 168, 171], HMMs and variants [12, 13, 42, 47, 82, 93, 103, 143, 157, 161, 174], principal component analysis

(PCA) and multiple discriminant analysis (MDA) [20, 30, 35, 68, 85], decision trees [62, 61], nearest-neighbour matching [87], image template matching [56, 147], correlation [150], rule-based methods [63, 74, 75, 80, 100, 146], and the semi-continuous dynamic Gaussian mixture model [167].



(a) (b)

Figure 2.1: Schemes for integration of component-level results: (a) System block diagram of a two-stage classification scheme by Vamplew [153], (b) Parallel HMMs where tokens are passed independently in the left and right hand channels, and combined in the word end nodes (E). S denotes word start nodes [158].

Since the approach taken in this work is to integrate simultaneous components, we will examine schemes for doing this in greater detail in the next section. Table 2.1 summarizes some of these schemes which are divided into approaches using direct-measure devices and cameras for acquiring hand gesture data. In vision-based methods single camera, stereo cameras or orthogonally placed cameras are used for image/video acquisition. Direct-measure (glove-based) devices for acquiring hand gesture data, consist of trackers that report position and orientation in 3D and gloves that measure the flexure and possibly abduction of finger joints using various types of sensors: optical (VPL Dataglove [1, 178]), resistor-based (Virtex Cyberglove [4]), magnetic (TUB-SensorGlove [65]), or accelerometers (AcceleGlove [62]). Electromagnetic trackers report 3D position and orientation (Polhemus 3Space [2], Ascension's Flock of Birds), ultrasonic trackers report 3D position only (PowerGlove [3, 146]), and the accelerometer-type tracker of the TUB-SensorGlove reports 3D orientation/acceleration. Hernandez-Rebollar et al. [61] recently experimented with a two-link mechanical arm skeleton fitted with an accelerometer and resistive angular sensors to measure rotation and flexion of the arm and forearm.

Sections 2.1.2 and 2.1.3 examine works that deal with two of the issues in SL recognition that are addressed in this thesis, grammatical processes and signer adaptation.

Table 2.1: Selected sign recognition systems using component-level classification.

Direct-measure device approaches

	Sign	No. of component categories				I/C	S/B	Classification of component-	Subjects		Rec.
Works	vocab.	HS	Mov.	Loc.	Orien.			level & sign-level	Train	Test	rate%
Hernandez[61]	176 ASL	42	7	11	6	I	S	Decision trees & Dict. lookup	17	1	94
Liang[93]	72-250	51	8	2	6	I	S	HMM & Dynamic prog.	1	1	78.4
	TWL					C^1	S	w/ stochastic grammar	1	1	84.7
Sagawa[136]	17 JSL	not stated				C^2	B	Matching with probs. & Dict. lookup	-	-	86.6
Su[145]	90 TWL	34	none	none	none	I	B	Hyperrectangular Composite NNs	2	2	94.1
								& Sum of similarity meas.	2	2*	91.2
Vamplew[154]	52 Auslan	30	13	19	15	I	S	Multilayer perceptron NN	7	4	94.2
								& Nearest-neighbour lookup	7	3*	85.3
Vogler[157]	22 ASL	71	140		none	C^3	B	HMM & Parallel HMM	1	1	95.5

Vision-based approaches

	Sign	No. of component categories				I/C	S/B	Features extracted	Classification	Subjects		Rec.
Works	vocab.	HS	Mov.	Loc.	Orien.				comp. & sign	Train	Test	rate%
Holden[64]	22 signs‡	22	none	none	none	I	S	3D hand model, 21 DOF	Fuzzy rules w/ adaptive distribs.	1	1	95
Imagawa[68]	33 JSL	?	3		?	I	B	2D segmented hand	PCA+clustering & Dict. lookup	6	6	72-94
Tamura[148]	10 JSL	2	5	5	?	I	S	2D hand contour, position	Rule-based & Dict.lookup	?	?	45

I/C:**I**solated or **C**ontinuous signing. S/B:**S**ingle hand or **B**oth hands. *Testing on unregistered signer(s).
‡Included Auslan signs and artificial signs. [1]Test set was 345 sentences averaging 4.7 words in length. [2]100 sentences were used as test set.[3]Test set of 99 sentences,each consisting of 2-7 signs,in unconstrained (but grammatical) word order.

2.1.1 Schemes for integrating component-level results

A common approach to integrate component-level results is to specify using domain knowledge, the categories of handshape, hand orientation, hand location and movement path shape that make up each sign in the vocabulary, forming a lexicon of sign definitions. Classifying the sign label from component-level results is then performed by comparing the ideal lexicon categories with the corresponding recognized components [61, 68, 80, 136, 145, 148, 154]. Various methods of performing this matching operation have been implemented; for example, Vamplew and Adams [154] employed a nearest-neighbour algorithm with a heuristic distance measure for matching sign candidates. In Sagawa and Takeuchi [136] the dictionary entries defined the mean and variance (which were learned from training examples) of handshape, orientation and motion type attributes as well as the degree of overlap in the timing of these components. Candidate signs were then given a probability score based on the actual values of the component attributes in the input gesture data. In Su [145] work on Taiwanese Sign Language (TWL), scoring was based on an accumulated similarity measure of input handshape data from the first and last 10 sample vectors of a sign. A major assumption was that signs can be distinguished based on just the starting and ending handshapes. This assumption is in fact only valid for some and not all signs. Liang and Ouhyoung [93] classified all four sign components using HMMs. Classification at the sign and sentence level was then accomplished using dynamic programming, taking into account the probability of the handshape, location, orientation and movement components according to dictionary definitions as well as unigram and bigram probabilities of the signs.

Methods based on HMMs include Gao et al. [47], where HMMs model individual signs while observations of the HMM states correspond to component-level labels for position, orientation and handshape, which were classified by multilayer perceptrons (MLPs). Vogler [157] proposed the parallel HMM algorithm to model sign components and recognize continuous signing in sentences. The right hand's shape, movement and location, along with left hand's movement and location were represented by separate HMM channels which were trained with relevant data and features. For recognition, individual HMM networks were built in each channel and a modified Viterbi decoding algorithm searched through all the networks in parallel. Path probabilities from each network that went through the same sequence of signs were combined (Figure 2.1(b)). Tanibata et al. [149] proposed a similar scheme where output probabilities from HMMs which model the right and left hand's gesture data were multiplied together for isolated sign recognition.

Waldron and Kim [159] combined component-level results (from handshape, hand location, orientation and movement type classification) with NNs, by experimenting with MLPs as well as Kohonen's self-organizing maps (SOM). The SOM performed slightly worse than the MLP (83% vs 86% sign recognition accuracy), but it was possible to relabel the map to recognize new signs without requiring additional training data (experimental results were given for relabelling to accomodate two new signs). In an adaptive fuzzy expert system ([29]) by Holden [64], signs were classified based on start and end handshapes and finger motion, using triangular fuzzy membership functions, whose parameters were found from training data.

An advantage of decoupling component-level and sign-level classification is that fewer classes would need to be distinguished at the component-level. This conforms with the findings of sign linguists that there are a small, limited number of categories in each of the sign components which can be combined to form a large number of signs. For example, in Liang and Ouhyoung [93], the most number of classes at the component-level was 51 categories (for handshape), which is smaller than the 71 to 250 signs that were recognized. In general, this approach enables the component-level classifiers to be simpler, with fewer parameters to be learned, due to the fewer number of classes to be distinguished and the reduced input dimensions (since only the relevant component features are input to each classifier). In the works where sign-level classification was based on a lexicon of sign definitions, training data only at the component-level classification was required, and not at the whole-sign level [61, 80, 93, 145, 148, 154, 157]. Furthermore, new signs can be recognized without retraining the component-level classifiers, if they cover all categories of components that may appear in signs. For example, the system of

Hernandez-Rebollar et al. [61] which was trained to classify 30 signs, was expanded to classify 176 new signs by just adding their descriptions into the lexicon. This system was however only used for classifying isolated signs.

Our approach to integrating component-level results and modelling multiple simultaneous components differs in two major ways. Firstly, a dictionary-definition of signs is not assumed, i.e., the relationship between sign and component-level results is not taken to be deterministic, but probabilistic, where the probability parameters are learned from training data. In contrast, most of the above works employ a dictionary-definition of the sign lexicon. Waldron and Kim [159] is an exception, where component-level results are combined using a trained NN. However, the MLP and SOM architectures they used work best on isolated signs (indeed, the majority of previous work listed in Table 2.1 only deals with isolated signing). The NNs need all the component-level results to be input at the same time, and the learned parameters represent the relationship between the sign class output and all the component-level inputs. There is no way to extract the relationship between the sign and each of the component-level results. We show in Chapters 3 and 4 how the probabilistic approach can be applied to both isolated and continuous signing. The relationship between sign value and component-level results are represented by separate parameters for each component, and can be learned separately. Secondly, we interpret both, the lexical word meaning and the additional grammatical information that is simultaneously conveyed. In order to to do this, we model not only the basic sign parts that are conventionally modelled as sign components in previous work but also define additional movement attributes as sign components.

2.1.2 Grammatical processes

Generally there have been very few works that address grammatical processes that affect the spatial and temporal dimensions of sign appearance in systematic ways. HMMs, which have been applied successfully to lexical word recognition, are designed to tolerate variability in the timing of observation features which are the essence of temporal aspect inflections. The approach of mapping each isolated gesture sequence into a standard temporal length ([30, 171]) causes loss of information on the movement dynamics. The few works that address this important aspect of SL generally deal only with spatial variations. Sagawa and Takeuchi [134] deciphered the subject-object pairs of Japanese Sign Language (JSL) verbs in sentences by learning the (Gaussian) probability densities of various spatial parameters of the verb's movement from training examples and thereby calculated the probabilities of spatial parameters in test data. Six different sentences constructed from two verbs and three different subject-object pairs, were tested on the same signer that provided the training set, and were recognized with an average word accuracy of 93.4%. Braffort [24] proposed an architecture where HMMs were employed for classifying lexical words using all the features of the sign gesture (glove finger flexure values, tracker location and orientation), while directional verbs were classified by their movement trajectory alone and classifier signs were classified by their finger flexure values only. Sentences comprising seven signs from the three different categories were successfully recognized with 92-96% word accuracy. They further proposed a rule-based interpreter module to establish the spatial relationship between the recognized signs, by maintaining a record of the sign articulations around the signing space. Although they were not applied to sign recognition, Parametric HMMs were proposed in [165] to estimate parameters representing systematic variations such as the distance between hands in a two-handed gesture and movement direction in a pointing gesture. However, it is unclear whether the method is suitable for larger vocabularies that exhibit multiple simultaneous variations.

The works above only deal with a subset of possible spatial variations, with no straightforward extension to modelling systematic speed and timing variations. In Watanabe [162] however, both spatial size and speed information were extracted from two different musical conducting gestures with 90% success. This method first recognized the basic gesture using min/max points in the gesture trajectory, and then measured the change in hand centre-of-gravity between successive images to obtain gesture magnitude and speed information.

The main weaknesses of the works above is that firstly they recognize a very limited number of different signs. There are six different sign appearances in [134], seven signs in [24] and two different gestures in [162]. Secondly, except for Watanabe's work which is on musical gestures and not SL manual signing, the others tackle signs with spatial variations only. Thirdly, only one type of variation is expressed in the signs at any one time, and there are no instances of multiple simultaneous grammatical information being expressed through multiple simultaneous systematic variations.

2.1.3 Signer independence and signer adaptation

Analogous to speaker independence in speech recognition, an ideal sign recognition system would work “right out of the box”, giving good recognition accuracy for signers not represented in the training data set (unregistered signers). Sources of inter-person variations that could impact sign recognition accuracy include different personal signing styles, different sign usage due to geographical or social background [152], and fit of gloves in direct-measure device approaches. In this area, sign recognition lags far behind speech.

When the number of signers in the training set is small, results on test data from unregistered signers can be severely degraded. In Kadous [74], accuracy decreased from an average of 80% to 15% when the system that was trained on 4 signers was tested on an unregistered signer. In Assan and Grobel [7], accuracy for training on one signer and testing on a different signer was 51.9% compared to 92% when the same signer supplied both training and test data. Better results were obtained when data from more signers was used for training. In Vamplew and Adams [154], seven signers provided training data; test data from these same (registered) signers was recognized with 94.2% accuracy vs 85.3% accuracy for three unregistered signers. Fang et al. [42] trained a recognition system for continuous signing on five signers and obtained test data accuracy of 92.1% for these signers, compared to 85.0% for an unregistered signer. Classification accuracy for unregistered signers is also relatively good when only handshape is considered, perhaps due to less inter-person variation as compared to the other gesture components. For example, [57] and [145] reported 93-96% handshape classification accuracy for registered signers vs 85-91% accuracy for unregistered signers. Interestingly, Kong and Ranganath [85] showed similarly good results for classifying 3D movement trajectories. Test data from six unregistered signers were classified with 91.2% accuracy vs 99.7% for test data from 4 registered signers.

In speech recognition, performance for a new speaker can be improved by using a small amount of data from the new speaker to adapt a prior trained system without retraining the system from scratch. The equivalent area of signer adaptation is relatively new. The work in Chapter 3 is a first attempt at addressing this area.

2.2 Modelling signs with grammatical information

The central focus of this thesis is on processes where some parts of a sign are modulated to convey grammatical information that is additional to and does not alter the lexical meaning in the sign. These modulations affect the sign form primarily in the attributes of the sign's movement path i.e. the shape traced by the hand movement path in 3-dimensional space, and the path direction, size and speed. Of these, only the shape traced by the hand (path shape) is conventionally considered as one of the basic parts or building blocks of signs (see Section 1.1.1). The other path attributes are usually ignored in SL recognition work since they do not convey the lexical meaning of a sign. In contrast, our analysis of sign structure takes into account these attributes because they are information-bearing parts of the sign. We define as separate components, attributes which convey information, have a limited number of distinct values and which are combined to construct signs, regardless of whether these are signs that just convey lexical meaning or convey additional non-lexical meaning as well.

In our analysis of sign structure, the basic parts or components of a sign are defined as handshape, hand orientation, location, movement path shape, movement path direction, movement path size and movement path speed. We first look at handshape, hand orientation, and location.

- There is a limited number of distinct categories of handshape that are formed from finger configurations; these are called handshape values. The **handshape component** of one sign consists of one or more handshape values (in sequence).

- There is a limited number of distinct categories of direction/orientation that the hand/palm faces, called orientation values. The **orientation component** of one sign consists of one or more orientation values (in sequence).
- There is a limited number of distinct categories of location that the hand positions itself in 3-dimensional space, called location values. The **location component** of one sign consists of one or more location values (in sequence).

Generally, the precise way in which the finger configurations change from forming one handshape value to another is just a function of the handshape values at the start and end of the change. Thus it is not relevant as it carries no information. The same can be said about the direction/orientation that the hand/palm faces. However it is somewhat different for the 3-dimensional hand position in space. For example, if a sign has a different end location value as compared to the start location value, there can be multiple possibilities for the shape of the path traced in 3-dimensional space to get from the start to the end location. In fact this movement path has many attributes besides shape. We define below each of these attributes as a sign component:

- There is a limited number of distinct categories for shapes of the paths traced in 3-dimensional space, for example, straight-line, an arc, a circle etc; these are referred to as path shape values. The **path shape component** of one sign consists of one or more path shape values (in sequence). The modulations in the dimensions of *contouring* and *cyclicity* due to temporal aspect inflections affect the path shape value(s) of a sign (see Section 1.1.3). Modulations in contouring could change the path shape value from straight to circle, for example. Modulations in cyclicity could result in multiple path shape values (in sequence) for a sign, instead of a single value.

- There is a limited number of distinct categories of directions in which paths in 3-dimensional space can point towards – these are referred to as path direction values. The **path direction component** of one sign consists of one or more path direction values (in sequence). Directional verbs point to the subject and object of the verb by modulating the sign's path direction value(s) (see Section 1.1.2). Note that even though there is potentially an unlimited number of directions in which a directional verb can point, within a signing discourse the possible directions are limited by the position of the referents (either present or absent) that have been set up during the discourse, since the directional verbs would only point to these referents.

- There is a limited number of distinct categories of sizes for the paths in 3-dimensional space – these are referred to as path size values. The **path size component** of one sign consists of one or more path size values (in sequence). The modulations in size due to temporal aspect inflections affect the path size value(s) of a sign (see Section 1.1.3).

- There is a limited number of distinct categories of speeds for tracing the paths in 3-dimensional space – these are referred to as path speed values. The **path speed component** of one sign consists of one or more path speed values (in sequence). The modulations in the dimensions of *rate* and *evenness* due to temporal aspect inflections affect the path speed value(s) of a sign (see Section 1.1.3)[1].

Among the movement path attributes defined above, the path shape is generally pertinent for determining the lexical or word meaning of a sign, whereas modulations in the values of the other attributes are pertinent for determining grammatical information conveyed by the sign. Of course, the attributes of path direction, size and speed also exist in signs which only convey lexical meaning, without any additional grammatical information. The important point is that to convey grammatical information, values of these attributes are varied or modulated. So just as we need to recognize the value(s) of the handshape, orientation, location and path shape components (collectively called the **lexical components**) in order to determine the lexical meaning, we would need to recognize the value of the path

[1] Modulation in the dimensions of absence/presence of onset-offset *hold* and *tension* due to temporal aspect inflections is difficult to measure (for example, muscle tension in the hand and arm is not measurable from the 3-dimensional hand position sequence) and is not dealt with in our model.

direction, size and speed components in order to extract the grammatical information.

We note that Liddell's [95] definition of features in movement segments does include some of the movement path attributes mentioned above (see Section 1.1.1). For example, the 'path contour' in his analysis is similar to path shape (as defined above), similarly qualities like 'shortening' and 'acceleration' are similar to path speed (as defined above), and path 'reduction' and 'enlargement' are similar to path size (as defined above). The model we propose however differs in two ways. Firstly, we consider movement path attributes to be simultaneous components on equal par with handshape, orientation and location components, and not as attributes of separate movement segments. Secondly, our model is non-commital with regards to segmental structure in signs, i.e. with regards to the movement and hold segments as defined by Liddell. We consider a sign as consisting of synchronized sequences of distinct values in each component. The sequences are synchronized at the start and end of the sign, since each component is expressing the same sign at the same time. There may or may not be sequential segments within a sign but in any case there is no requirement for the component sequences to be synchronized at segment boundaries, the only requirement is synchronization at the sign boundaries. Liddell's definition implies synchronization between components not only at sign boundaries but also at sub-sign segments boundaries.

2.3 Overview of approach

The information conveyed by a sign includes lexical/word meaning and possibly multiple and simultaneously expressed categories of grammatical information. This information is conveyed through the physical appearance of the sign, with the grammatical information most significantly expressed in movement path attributes that are not conventionally modelled as basic sign parts or components. Previous work has modeled the sign components that identify lexical meaning as simultaneous and independent components (refer Section 2.1.1). These sign **lexical components** are handshape, orientation, location and movement path shape. This approach is generalized by modelling not only the lexical components, but also the various temporal and spatial movement attributes that exhibit systematic variation (specifically movement path direction, size and speed), as independent information-carrying components, with distinct "primes" or values that are classified from separate feature sets (refer Section 2.2). There are a limited number of these distinct values in each component and they combine to produce a large number of different signs. Thus data from multiple signs can be pooled together for training the component-level classifiers.

The goal is to build models whose structure reflects the effect of lexical and grammatical information conveyed in the sign, on each of the components, train the model, and then use the trained model to infer the information conveyed in a sign or sign sequence through observing feature data streams in each of the components. We use a probabilistic framework for the models, viz., Bayesian networks (BNs) for isolated gestures (Chapter 3) and dynamic Bayesian networks (DBNs) for continuous signs (Chapter 4). The model structure explicitly represents our domain knowledge of the lexical and grammatical structure of sign language and the assumption of independent components. The advantage of this simplifying assumption is that we need never model the interaction between all the components in a sign, thereby greatly reducing the number of model parameters. These parameters numerically define the probabilistic relationships between the information conveyed through a sign and the sign component values, and are learned from training data, rather than assuming them to be deterministic, and specifying their values.

The probabilistic approach for modelling sign to component dependencies is different from most previous work for combining component-level results which commonly assume a dictionary definition or deterministic dependencies between sign and components. The probabilistic approach does not require data additional to that required for training component-level classifiers and can improve on component-level classifier accuracies. There are commonalities across signs in their effects on sign components. So even though the sign to component dependencies are numerically defined and need to be learned, the commonalities can be exploited to reduce the model parameters required by allowing signs to share parameters.

Chapter 3

Recognition of isolated gestures with Bayesian networks

This chapter describes a framework for recognizing isolated gestures displaying systematic variations in temporal and spatial movement attributes along with experiments using digital video data gestures. The gesture vocabulary is novel and defined to have a similar structure as signs carrying grammatical information. The gestures convey both basic meaning (which is identified from the values of the gestures' lexical components) and additional meaning (equivalent to inflections in signs) which modulate movement attributes in systematic ways. The lexical components and movement attributes are considered to be independent components of the gesture, each with a limited number of categories or classes. The approach here is to define the distinct classes in each component and train component-level classifiers. A Bayesian network (BN) is then used to combine results from the trained component-level classifiers, and infer basic meaning and inflections in the gesture. The BN parameters are learned from the same training data that is used to train the component-level classifiers.

Although the main focus of this thesis is the analysis and recognition of inflectional processes, in Section 3.2 we also consider another oft-neglected issue in sign language (SL) recognition, that of signer adaptation, and propose a framework for adapting a trained system to yield improved performance on a new signer. Experimental results are likewise reported for the implementation of this signer adaptation scheme.

3.1 Overview of proposed framework and experimental setup

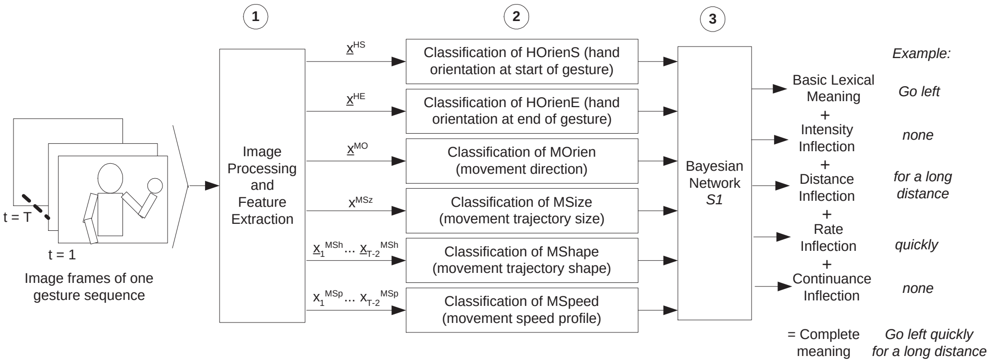

1
2
3
t = T
t = 1
Image frames of one gesture sequence
Image Processing and Feature Extraction
$\underline{x}^{HS}$
$\underline{x}^{HE}$
$\underline{x}^{MO}$
x^{MSz}
$\underline{x}_1^{MSh} \dots \underline{x}_{T-2}^{MSh}$
$x_1^{MSp} \dots x_{T-2}^{MSp}$
Classification of HOrienS (hand orientation at start of gesture)
Classification of HOrienE (hand orientation at end of gesture)
Classification of MOrien (movement direction)
Classification of MSize (movement trajectory size)
Classification of MShape (movement trajectory shape)
Classification of MSpeed (movement speed profile)
Bayesian Network S1
Basic Lexical Meaning
+
Intensity Inflection
+
Distance Inflection
+
Rate Inflection
+
Continuance Inflection
= Complete meaning
Example:
Go left
none
for a long distance
quickly
none
Go left quickly for a long distance

Figure 3.1: System block digram showing: (1) image processing and feature extraction, (2) component-level classification, and (3) Bayesian network, *S1*, for inferring basic meaning and inflections. Example final output from the system is shown on the right.

The block diagram in Figure 3.1 shows an overview of the processing steps in the proposed system. In Step 1, the input gesture video is processed to extract features that are appropriate to classify (i) the hand orientation at the start and (ii) end of the gesture, (iii) the movement path orientation/direction and (iv) size (also based on information at start and end of gesture), and (v) the movement trajectory/path shape and (vi) speed profile (based on information obtained throughout the gesture sequence). As our focus here is on developing a classification framework for interpreting inflections in signing, we simplified the imaging conditions and image processing operations. The test subjects performed the gestures while wearing black gloves and a white long-sleeved shirt, and a white board was used as background. In Step 2, six trained classifiers independently categorize these features for input to a BN. The BN structure is developed using domain knowledge as described in Section 3.1.4. The conditional probability tables (CPTs) for the network are learned from training data. After training, the complete sign meaning including inflections can be inferred. This is shown as the output of Step 3 in Figure 3.1.

In the next section, we describe the gesture vocabulary used in the experiments before passing on to Steps 1, 2 and 3 of the block diagram.

3.1.1 Gesture vocabulary

We used a simulated vocabulary with 6 basic meanings (“Go left”, “Go right”, “Good”, “Bad”, “Bright”, “Dark”), and 5 possible inflections (“very”, “continuously”, “for a long time”,“quickly”, “for a long distance”,) which together can form 20 distinct gestures as shown in Table 3.1 and Figure 3.2. This includes inflections that modify the movement both temporally and spatially; and is a larger vocabulary than that used in previous related work on recognizing inflected signs. The vocabulary is designed to have fewer ambiguities in the 2-dimensional image plane, while adhering to the general principle of how basic lexical meaning and inflections are combined in ASL. This allows us to keep the image processing part of the scheme simple, and focus on classifying the movements. The basic meaning of a gesture is represented by the pointing direction of the thumb (hand orientation), movement trajectory/path shape, and movement path direction/orientation; these are equivalent to lexical components in our vocabulary. The inflections “very”, “continuously” and “for a long time”, are characterized by movement variations as described in Examples 2 and 3 and Figure 1.5 of Section 1.1.3: the modulation adding the meaning “very” affects the movement characteristics of tension, hold, rate and size; the modulations which add the meanings “continuously” and “for a long time” affect the rate, evenness, contouring and cyclicity of the movement. On the other hand, the modulations which add the meaning “quickly” and “for a long distance" affect only the rate and size of movement, respectively, and can co-occur. Since different types of inflections are associated with different basic meanings (for example "very go left" does not make any sense), the BN structure must take this into account and interpret the movement manner differently depending on the basic gesture meaning.

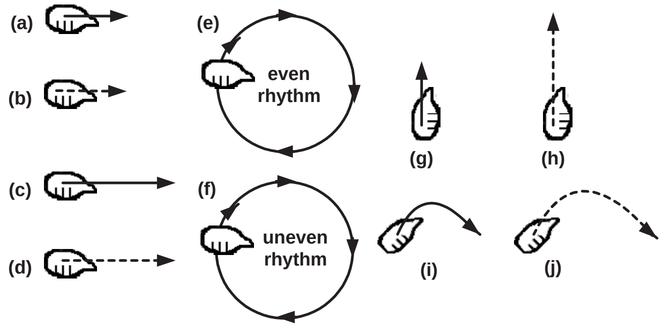


Figure 3.2: Ten of the possible combinations of basic meaning and inflections: (a) "Go left", (b) "Go left quickly", (c) "Go left for a long distance", (d) "Go left quickly for a long distance", (e) "Go left continuously", (f) "Go left for a long time", (g) "Good", (h) "Very good", (i) "Bright", (j) "Very bright". "Go right", "Dark" and "Bad" gestures are flipped versions of "Go left", "Bright" and "Good" respectively. (Solid (dotted) lines denote medium (fast) speed).

3.1.2 Step 1: image processing and feature extraction

An NTSC digital color video camera was used to capture 320x240 24-bit color image sequences at frame rates of 5-15fps. The videos were then manually segmented in time to obtain isolated gesture actions. In Step 1 of Figure 3.1, the hand is first automatically segmented out in each image, as shown in Figure 3.3, by thresholding, based on color and frame differences. The resulting binary image is used to obtain the hand centroid and axis of least inertia (determined by the major

Table 3.1: Complete list of sign vocabulary (20 distinct combined meanings)

Go left
Go right
Good
Bad
Bright
Dark
{Go left, Go right} for a long distance
{Go left, Go right} quickly
{Go left, Go right} quickly for a long distance
{Go left, Go right} continuously
{Go left, Go right} for a long time
Very {Good, Bad, Bright, Dark}

axis of the bounding ellipse of the hand [142]). The angle of this axis, ϕ, and 3rd order moments are used to distinguish between orientations of, for example, down vs up. We define six gesture components, and for classifying each component into a distinct category, we extract one of the following feature vectors:

- $\underline{x}^{HS} = [sin\phi_1, cos\phi_1]^T$ where ϕ_1 is the angle of the axis of least inertia in the first gesture frame[1]. This is used for classifying hand orientation at the start of the gesture, in the **HOrienS** component.

- $\underline{x}^{HE} = [sin\phi_T, cos\phi_T]^T$ where ϕ_T is the angle of the axis of least inertia in the last gesture frame. This is used for classifying hand orientation at the end of the gesture, in the **HOrienE** component.

[1] Measuring features from a single frame, instead of averaging over a small temporal window could result in more susceptibility to noise. However due to the low frame rate of the video captured, the latter approach would likely "smudge" the feature measurements.

- $\underline{\mathrm{x}}^{MO} = [sin\alpha, cos\alpha]^T$ where α is the angle of the straight line between the hand centroid in the first and last frames, used for categorizing movement orientation/direction in the **MOrien** component.

- x^{MSz} is the length of the straight line between the hand centroid in the first and last frames, used for categorizing movement path size in the **MSize** component.

- $\underline{\mathrm{x}}_t^{MSh} = [sin\theta_t, cos\theta_t]^T$, $t = 1, \ldots, T-2$ where θ_t is the change in the motion vector angle in successive video frames, defined as in Figure 3.4. This sequence of features is extracted from all but the last two of the T frames in one gesture action, and is used to categorize movement path shape and cyclicity in the **MShape** component.

- x_t^{MSp}, $t = 1, \ldots, T-2$ is the difference of the motion vector magnitudes in successive image frames (Figure 3.4). This sequence of features is used to categorize the speed profile of the movement in the **MSpeed** component, accounting for rate and evenness of movement.

3.1.3 Step 2: component-level classification

The features obtained in Step 1 are categorized by component-level classifiers.

The input to gesture components, HOrienS, HOrienE, MSize and MOrien, are static features. For classification in each of these gesture components, we assume

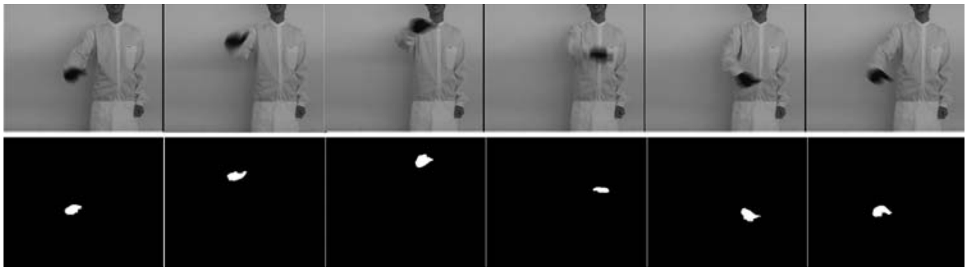


Figure 3.3: Example image sequence of "Go left continuously" and corresponding thresholded images.

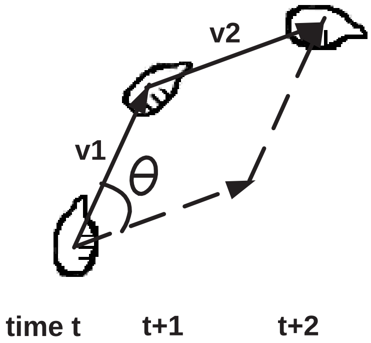


Figure 3.4: Illustration of change in motion vector angles (θ) and change in motion magnitude ($\mathrm{x}_t^{MSp} = ||\vec{v2}|| - ||\vec{v1}||$)

class-conditional Gaussian mixture densities (with 2 to 10 mixture components) for the relevant features of that gesture component. The parameters of these densities are computed using the maximum likelihood (ML) criterion, and estimated using the Expectation-Maximization (EM) algorithm. For example, Gaussian mixtures are estimated for each of the six categories of HOrienS viz; *Left*, *Right*, *Up*, *Down*, *Diagonal-Left*, and *Diagonal-Right*. Similarly, six categories are defined for components HOrienE and MOrien, while three categories are defined for MSize. Subsequently the trained component-level classifier yields the class/category with the highest likelihood for a given input feature vector. This is a generative approach to

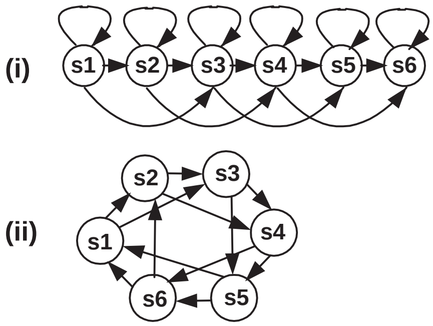


Figure 3.5: State transition diagrams for hidden Markov models.

classification. A discriminative graphical model approach or other discriminative classifiers like neural networks are also possible and may require fewer parameters to train [70]. However, we did not make comparisons between these alternative approaches as the main focus of the work in this chapter is on evaluating the feasibility of using a Bayesian Network to combine component-level classification results and not on evaluating different types of component-level classifiers. Learning class-conditional densities also made it easier to make a comparison with the approach in a previous work ([134]) which directly multiplied the probability scores of component features (see Section 3.3.1).

For the MShape component, where a time sequence of data points is classified, we train, using ML estimation, one HMM for each of the 5 categories, *Straight*, *Left-Arc*, *Right-Arc*, *Counter-Clockwise-Circle* and *Clockwise-Circle*. A new test sequence is then classified according to the HMM which gives the highest likelihood for the time sequence of data points. A 6-state left-right (Bakis) HMM structure (Figure 3.5(i)) is used for all categories except *Counter-Clockwise-Circle* and *Clockwise-Circle* which have no self-transitions and have an additional loop-back state transition to the first state to account for the multiple cycles in the circular paths (Figure 3.5(ii)). The state output densities are single Gaussians. The EM algorithm is used for training and is terminated when the percentage increase in log-likelihood between iterations falls below a threshold. Similarly, one HMM is trained for each of the 4 categories in the MSpeed component — the structure in Figure 3.5(i) is used for categories *Medium* and *Fast*, while the structure in Figure 3.5(ii) is used for *Even* and *Uneven*.

We choose observation features for the MShape HMMs that, as far as possible, are not influenced by the size and speed of the gesture movement. The chosen features — the change in the angle of motion vectors — do not include explicit measurements of hand position and motion vector magnitude so that each MShape HMM can be trained with data from gestures with different movement sizes and speed profiles without incurring a large variation in the observation features. Similarly, changes in the speed are the observation features for the MSpeed HMMs, and do not include explicit information on hand position and movement path shape.

3.1.4 Step 3: BN for inferring basic meaning and inflections

The final stage of the system is implemented with a Bayesian network (BN) which is a directed acyclic graph consisting of a set of nodes representing random variables, $\mathbf{Y} = Y_1, \ldots, Y_n$, and directed edges representing dependencies among the nodes [59]. In the graph, absence of edges implies conditional independence, i.e. a node is independent of its non-descendants, given its parents. The conditional independencies encoded in the graph allow the joint distribution of the set of random variables to be factored as a product of local conditional probabilities: $P(Y_1, \ldots, Y_n) = \prod_{i=1}^{n} P(Y_i \mid \mathbf{Pa}_{Y_i})$, where $\mathbf{Pa}_{Y_i}$ is the set of parents of random variable Y_i. Although the network structure which encodes the conditional independence relationships can be learned from training data, in many applications, the structure is manually defined using domain knowledge of the problem. As such, training of the network consists of learning the network parameters, $\boldsymbol{\theta}$, which are the numerical values of the local conditional probabilities, from training data, D. The training data, $D = \{\mathbf{y}[1], \ldots, \mathbf{y}[N]\}$, is assumed to be a random sample from the joint probability distribution of $\mathbf{Y}$. Network parameters, $\boldsymbol{\theta}$, can be learned using either ML estimation or Bayesian estimation if all the node values are known at training time. After training the network allows inferring the probabilities of query nodes given the observed values of evidence nodes.

Though gestures can be viewed as being described through a set of rules, a BN for probabilistic inferencing is prefered to rule-based deduction. This is in view of the inherent uncertainties which can manifest themselves through component-level classifier errors, which arise from inaccuracies and noise in feature extraction, and inter-person or even intra-person variations between individual gesture performances. BNs can account for these uncertainties, and are therefore useful to represent gestures and their inflections.

We define five query nodes in the BN. The BasicMeaning node represents six possible basic gesture meanings ("Go left", "Go right", "Good", "Bad", "Bright", "Dark"), while the other four nodes represent the absence or presence of inflections. These are Intensity (none, "very"), Distance (none, "for a long distance"), Rate (none, "quickly") and Continuance (none, "continuously", "for a long time"). The observation nodes represent the six gesture components, HOrienS, HOrienE, MShape, MOrien, MSize, and MSpeed. The possible values of these nodes, L^{HS}, L^{HE}, L^{MSh}, L^{MO}, L^{MSz} and L^{MSp} are the discrete categories of each of the components. Here the network structure is defined using prior knowledge, and the rationale for the precise structure is discussed in the following.

The lexical components that represent the gesture's basic meaning are: (i) hand orientation at start, and (ii) end of gesture; (iii) movement path shape; and (iv) movement path orientation. Given the class label of BasicMeaning, the lexical component categories are assumed to be mutually independent, and this conditional independence relationship is represented by the network in Figure 3.6(a). To deal

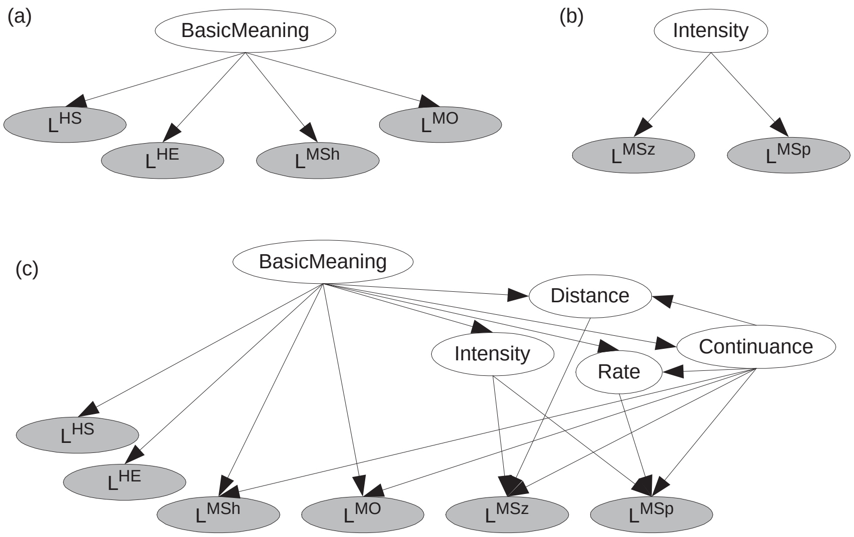


Figure 3.6: (a) Conditional independence of lexical components, (b) causal dependence between movement attributes and Intensity node, (c) *S1* network models the causal relationship between basic gesture meaning, inflections, lexical components and movement attributes.

with inflections, we note, for example, that a gesture with the inflection “very” has larger movement size and speed as compared to the uninflected gesture, while the lexical gesture components of hand orientation, movement shape and movement orientation are unaffected. We can conceptualize this as a “causal” relationship between the Intensity inflection node and the MSize and MSpeed nodes (represented by the network in Figure 3.6(b)). Similar causal relationships can be represented by edges between the other inflection nodes Distance, Rate and Continuance, and the relevant gesture components.

Since different types of inflections are associated with different gestures, edges are added from BasicMeaning to Intensity, Distance, Rate and Continuance. Furthermore, edges from Continuance to the Rate and Distance nodes take into account how the inflections, “continuously” and “for a long time”, cannot co-occur with the inflections “quickly” or “for a long distance”. By taking into account these considerations, and the causal relationships represented by the networks in Figure 3.6(a) and Figure 3.6(b), we arrive at the network structure *S1* (Figure 3.6(c)) which encodes the causal relationships between the gesture’s basic meaning and inflections, and the component category labels output by component-level classifiers.

3.1.5 Training the Bayesian network

The network parameters, $\boldsymbol{\theta}$, for *S1*, can now be learned from training data, D, using ML estimation. Due to network factorization, the likelihood, $P(D|\boldsymbol{\theta})$, decomposes according to the structure of the network,

$$\begin{aligned}
P(D|\boldsymbol{\theta}) &= P(\mathbf{y}[1], \ldots, \mathbf{y}[N]|\,\boldsymbol{\theta}) \\
&= \prod_{l=1}^{N} P(\mathbf{y}[l]|\,\boldsymbol{\theta}) \\
&= \prod_{l=1}^{N} \prod_{i=1}^{n} P(Y_i = y_i[l] \mid \mathbf{Pa}_{Y_i} = \mathbf{pa}_{Y_i}[l], \boldsymbol{\theta}_i) \\
&= \prod_{i=1}^{n} \left\{ \prod_{l=1}^{N} P(Y_i = y_i[l] \mid \mathbf{Pa}_{Y_i} = \mathbf{pa}_{Y_i}[l], \boldsymbol{\theta}_i) \right\} \qquad (3.1)
\end{aligned}$$

where $\boldsymbol{\theta}_i$ denotes the parameters of the local distribution function $P(Y_i \mid \mathbf{Pa}_{Y_i})$.

These parameters can be estimated independently, since, from (3.1),

$$\begin{aligned}
\hat{\boldsymbol{\theta}} &= \operatorname*{argmax}_{\boldsymbol{\theta}} P(D|\boldsymbol{\theta}) \\
&= \operatorname*{argmax}_{\boldsymbol{\theta}} \log P(D|\boldsymbol{\theta}) \\
&= \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^{n} \left\{ \sum_{l=1}^{N} \log P(Y_i = y_i[l] \mid \mathbf{Pa}_{Y_i} = \mathbf{pa}_{Y_i}[l], \boldsymbol{\theta}_i) \right\} \qquad (3.2)
\end{aligned}$$

Hence, we have independent estimation problems for each $\boldsymbol{\theta}_i$,

$$\hat{\boldsymbol{\theta}}_i = \operatorname*{argmax}_{\boldsymbol{\theta_i}} \sum_{l=1}^{N} \log P(Y_i = y_i[l] \mid \mathbf{Pa}_{Y_i} = \mathbf{pa}_{Y_i}[l], \boldsymbol{\theta}_i) \qquad (3.3)$$

In a network such as *S1* where all the variables Y_i are discrete, with possible values, $k = 1, \ldots, r_i$, the local distribution function for Y_i is a collection of distinct multinomial distributions, one distribution for each configuration of its parents $\mathbf{Pa}_{Y_i}$. So $\boldsymbol{\theta}_i = (\boldsymbol{\theta}_{i1}, \ldots, \boldsymbol{\theta}_{iq_i})$, where the possible configurations of the parents are $j = 1, \ldots, q_i$. For each such configuration j, the vector of parameters of the multinomial is $\boldsymbol{\theta}_{ij} = (\theta_{ij1}, \ldots, \theta_{ijr_i})$, where $\theta_{ijk} \triangleq P(Y_i = k|\mathbf{Pa}_{Y_i} = j)$, for $k = 1, \ldots, r_i$. The parameter vectors, $\boldsymbol{\theta}_{ij}$ (for $i = 1, \ldots, n$ and $j = 1, \ldots, q_i$), are assumed to be mutually independent, hence can be estimated independently. The ML estimation of the parameters of a multinomial distribution are the sample proportions [59],

$$\hat{\theta}_{ijk} = \frac{N_{ijk}}{\sum_{m=1}^{r_i} N_{ijm}} \tag{3.4}$$

where N_{ijk} is the number of times $Y_i = k$ and $\mathbf{Pa}_{Y_i} = j$ occur in the observation data set.

In the next section we describe a scheme developed to adapt both the trained component-level classifiers and the trained *S1* network to yield good results for new signers. In Section 3.3, we show experimental results of trained *S1* networks which combine the results of component-level classifiers for inferring class labels for basic gesture meaning and inflections.

3.2 Signer adaptation scheme

We now describe a method for adapting a trained Bayesian network-based multiple signer system to recognize gestures performed by a new test subject, using only a small amount of adaptation data from the new subject. The signer adaptation scheme separately adapts the component-level classifiers and the network *S1*.

3.2.1 Adaptation of component-level classifiers

In Section 3.1.3, the approach to component-level classification was to train, using ML estimation, in each component, a set of models whose parameters best explained training examples for the known category. However, if the ML approach is followed to train a set of models on gestures performed by a new person, when only a small amount of adaptation data is available, model parameter estimates will not be robust. Our approach for adapting to a new person is to use maximum a posteriori (MAP) estimation, which is one of the main speaker adaptation schemes in speech recognition systems that are based on continuous density HMMs [49, 91].

Each of the HOrienS, HOrienE, MOrien and MSize components is characterized by class-conditional Gaussian mixtures. For a particular component, and a particular class/category in that component, the joint probability distribution function (p.d.f) of T independent, identically-distributed (i.i.d) observations $\underline{x}_t$ drawn from that class is,

$$\begin{aligned} p(\mathbf{x}|\boldsymbol{\theta}) &= \prod_{t=1}^{T} p(\underline{x}_t|\boldsymbol{\theta}) \\ &= \prod_{t=1}^{T} \sum_{i=1}^{M} \omega_i \mathcal{N}(\underline{x}_t|\underline{\mu}_i, \Sigma_i) \end{aligned} \tag{3.5}$$

where $\mathbf{x} = (\underline{x}_1, \ldots, \underline{x}_T)$, $\boldsymbol{\theta} = (\omega_1, \ldots, \omega_M, \underline{\mu}_1, \ldots, \underline{\mu}_M, \Sigma_1, \ldots, \Sigma_M)$, ω_i are mixture weights, $\mathcal{N}(\underline{x}_t|\underline{\mu}_i, \Sigma_i)$ are Gaussian distributions with mean $\underline{\mu}_i$ and covariance matrix Σ_i, and all the class-conditional mixtures are assumed to have M components to simplify illustration. We can formulate $p(\underline{x}_t|\boldsymbol{\theta})$ as a marginal probability by introducing hidden variable l_t — the unobserved label of the mixture component. Let the mixture weight for the i-th component, ω_i, be the probability that l_t takes on the value i, i.e. $\omega_i \triangleq P(l_t = i)$, and let $p(\underline{x}_t|l_t = i)$ be given by the mixture component density, $\mathcal{N}(\underline{x}_t|\underline{\mu}_i, \Sigma_i)$. We then obtain $p(\underline{x}_t|\boldsymbol{\theta})$ by summing over i [70],

$$\begin{aligned} p(\underline{x}_t|\boldsymbol{\theta}) &= \sum_{i=1}^{M} p(\underline{x}_t, l_t = i|\boldsymbol{\theta}) \\ &= \sum_{i=1}^{M} P(l_t = i|\boldsymbol{\theta}) p(\underline{x}_t|l_t = i, \boldsymbol{\theta}) \\ &= \sum_{i=1}^{M} \omega_i \mathcal{N}(\underline{x}_t|\underline{\mu}_i, \Sigma_i) \end{aligned} \tag{3.6}$$

which gives us the mixture model of (3.5).

In the MAP estimation of the model parameters, prior knowledge about the parameters in the form of a prior distribution for $\boldsymbol{\theta}$, is used in addition to the adaptation data $\mathbf{x}$ from the new person to provide a more robust estimate (than if $\mathbf{x}$ alone was utilized as in ML estimation). With the introduction of hidden variables, $\mathbf{l} = (l_1, \ldots, l_T)$, we can use the EM algorithm to iteratively refine the model parameters with the goal of maximizing the posterior probability of $\boldsymbol{\theta}$, given $\mathbf{x}$. This is equivalent to maximizing the logarithm of the joint distribution of $\mathbf{x}$ and $\boldsymbol{\theta}$ [33]:

$$\begin{aligned} \hat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log p(\mathbf{x}, \boldsymbol{\theta}) \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log \left\{ \sum_{\mathbf{l}} p(\mathbf{x}, \mathbf{l}, \boldsymbol{\theta}) \right\}. \end{aligned} \tag{3.7}$$

In the M-step of the $(k+1)$-th iteration, the parameter values are re-estimated as [33],

$$\boldsymbol{\theta}^{k+1} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left\{ E_{P(\mathbf{l}|\mathbf{x},\boldsymbol{\theta}^k)}[\log p(\mathbf{x}, \mathbf{l}|\boldsymbol{\theta})] + \log p(\boldsymbol{\theta}) \right\}, \tag{3.8}$$

where the first term is the expected complete log likelihood, given the observed (adaptation) data $\mathbf{x}$, and the parameters estimated in the k-th iteration, $\boldsymbol{\theta}^k$. The second term is the log of the prior density assumed for the model parameters.

The scheme here is a simplified version of the method in [49], where the mixture component means, covariances, and weights are all adaptively refined using MAP estimation. In this investigation for signer adaptation, we only adapt the mixture component means $\boldsymbol{\mu} = (\underline{\mu}_1, \ldots, \underline{\mu}_M)$, while the other parameters of the Gaussian mixture, are assumed to be fixed and known. Mean adaptation seems a good place to start for a first attempt at signer adaptation because it reflects variations in the appearances of gestures across different test subjects. For example the angle at which test subjects held their hands, or the trajectory size in which they performed the same gesture type. Hence it is more pertinent for adaptation than mixture component covariances (which measure intra-person variations). By inspection of video data input in our experiments, we found that each test subject performed each type of gesture in a consistent manner (for eg. the appearances of different instances of “Go left for a long distance” for test subject A were consistent with one another). Hence there is relatively less difference between the intra-person variations of different signers. Another motivation for this approach is that it has been found in speech recognition (based on continuous density HMMs) that the most important speaker specific effect is related to the Gaussian means of state observation densities [166].

For each individual Gaussian mixture component, an appropriate distribution for modeling prior knowledge about the mean, $\underline{\mu}_i$, is a conjugate density, such as a Gaussian density,

$$p(\underline{\mu}_i) = \mathcal{N}(\underline{\mu}_i | \underline{\mu}_{io}, \Sigma_{io}). \tag{3.9}$$

where $\underline{\mu}_{io}$ represents our best guess for $\underline{\mu}_i$ and Σ_{io} represents our uncertainty about this guess [38]. Assuming independent parameters (i.e. the mean of a particular component is independent of the means of the other components), the joint prior density of the means is given by, $p(\boldsymbol{\mu}) = \prod_{i=1}^{M} p(\underline{\mu}_i)$.

Specializing the M-step equation (3.8) to adapt only the means, we get

$$\begin{aligned}
\boldsymbol{\mu}^{k+1} &= \operatorname*{argmax}_{\boldsymbol{\mu}} \left\{ E_{P(\mathbf{l}|\mathbf{x},\boldsymbol{\mu}^k)}[\sum_{t=1}^{T} \log p(\underline{x}_t, l_t | \underline{\mu}_1, \ldots, \underline{\mu}_M)] + \sum_{i=1}^{M} \log p(\underline{\mu}_i) \right\} \\
&= \operatorname*{argmax}_{\boldsymbol{\mu}} \left\{ \sum_{t=1}^{T} \sum_{i=1}^{M} P(l_t = i | \underline{x}_t, \boldsymbol{\mu}^k) \log p(\underline{x}_t, l_t = i | \underline{\mu}_i) + \sum_{i=1}^{M} \log p(\underline{\mu}_i) \right\} \\
&= \operatorname*{argmax}_{\boldsymbol{\mu}} \left\{ \sum_{i=1}^{M} \left[\sum_{t=1}^{T} P(l_t = i | \underline{x}_t, \boldsymbol{\mu}^k) \log p(\underline{x}_t, l_t = i | \underline{\mu}_i) + \log p(\underline{\mu}_i) \right] \right\}
\end{aligned} \tag{3.10}$$

Each of the terms within the outer summation can be maximized independently with respect to $\underline{\mu}_i$. Hence,

$$\begin{aligned}\underline{\mu}_i^{k+1} &= \underset{\underline{\mu}_i}{\operatorname{argmax}} \left\{ \sum_{t=1}^{T} P(l_t = i|\underline{x}_t, \boldsymbol{\mu}^k) \log p(\underline{x}_t, l_t = i|\underline{\mu}_i) + \log p(\underline{\mu}_i) \right\}, \\ &\quad \text{for } i = 1, \cdots, M \qquad (3.11)\end{aligned}$$

Defining,

$$\begin{aligned}\gamma_t(i)^k &= P(l_t = i|\underline{x}_t, \boldsymbol{\mu}^k) \\ &= P(l_t = i|\underline{x}_t, \underline{\mu}_1^k, \ldots, \underline{\mu}_M^k) \\ &= \frac{\omega_i \mathcal{N}(\underline{x}_t|\underline{\mu}_i^k, \Sigma_i)}{\sum_{m=1}^{M} \omega_m \mathcal{N}(\underline{x}_t|\underline{\mu}_m^k, \Sigma_m)} \qquad (3.12)\end{aligned}$$

we obtain, from equations (3.11),(3.6) and (3.9),

$$\begin{aligned}\underline{\mu}_i^{k+1} &= \underset{\underline{\mu}_i}{\operatorname{argmax}} \sum_{t=1}^{T} \gamma_t(i)^k \log \omega_i \mathcal{N}(\underline{x}_t|\underline{\mu}_i, \Sigma_i) + \log \mathcal{N}(\underline{\mu}_i|\underline{\mu}_{io}, \Sigma_{io}) \\ &= \underset{\underline{\mu}_i}{\operatorname{argmax}} \sum_{t=1}^{T} \gamma_t(i)^k \log \mathcal{N}(\underline{x}_t|\underline{\mu}_i, \Sigma_i) + \log \mathcal{N}(\underline{\mu}_i|\underline{\mu}_{io}, \Sigma_{io}) \qquad (3.13)\end{aligned}$$

since $\sum_{t=1}^{T} \gamma_t(i)^k \log \omega_i$ is independent of $\underline{\mu}_i$, and we have assumed that ω_i and Σ_i are fixed and known. Maximizing (3.13) is equivalent to maximizing its exponential,

$$\begin{aligned}&\exp\left[\sum_{t=1}^{T} \gamma_t(i)^k \log \mathcal{N}(\underline{x}_t|\underline{\mu}_i, \Sigma_i)\right] \exp\left[\log \mathcal{N}(\underline{\mu}_{io}, \Sigma_{io})\right] \\ &= K \exp\left[-\frac{1}{2}\left(\sum_{t=1}^{T} \gamma_t(i)^k (\underline{x}_t - \underline{\mu}_i)' \Sigma_i^{-1} (\underline{x}_t - \underline{\mu}_i) + (\underline{\mu}_i - \underline{\mu}_{io})' \Sigma_{io}^{-1} (\underline{\mu}_i - \underline{\mu}_{io})\right)\right] \\ &\qquad (3.14)\end{aligned}$$

where quantities that do not depend on μ_i have been absorbed into the constant K. The expression in (3.14) is an exponential function of $\underline{\mu}_i$ and is again a Gaussian density [38], $\mathcal{N}(\underline{\mu}_i|\underline{\hat{\mu}}_i, \hat{\Sigma}_i)$, where,

$$\begin{aligned} \underline{\hat{\mu}}_i &= \Sigma_{io}\left(\Sigma_{io} + \frac{1}{\sum_{t=1}^{T}\gamma_t(i)^k}\Sigma_i\right)^{-1}\frac{\sum_{t=1}^{T}\gamma_t(i)^k\underline{x}_t}{\sum_{t=1}^{T}\gamma_t(i)^k} \\ &\quad + \frac{1}{\sum_{t=1}^{T}\gamma_t(i)^k}\Sigma_i\left(\Sigma_{io} + \frac{1}{\sum_{t=1}^{T}\gamma_t(i)^k}\Sigma_i\right)^{-1}\underline{\mu}_{io} \qquad (3.15) \end{aligned}$$

$$\hat{\Sigma}_i = \Sigma_{io}\left(\Sigma_{io} + \frac{1}{\sum_{t=1}^{T}\gamma_t(i)^k}\Sigma_i\right)^{-1}\frac{1}{\sum_{t=1}^{T}\gamma_t(i)^k}\Sigma_i \qquad (3.16)$$

From (3.15),

$$\underline{\mu}_i^{k+1} = \underline{\hat{\mu}}_i \qquad (3.17)$$

since the mode of the Gaussian density, $\mathcal{N}(\underline{\mu}_i|\underline{\hat{\mu}}_i, \hat{\Sigma}_i)$, is at its mean $\underline{\hat{\mu}}_i$. This has an elegant interpretation as the weighted average of the ML estimate of the mixture component mean from adaptation data, $\sum_{t=1}^{T}\gamma_t(i)^k\underline{x}_t/\sum_{t=1}^{T}\gamma_t(i)^k$, and the prior density of the component mean, $\underline{\mu}_{io}$. From (3.15), in order to evaluate $\underline{\hat{\mu}}_i$, the parameters, $\underline{\mu}_{io}$ and Σ_{io}, of the prior density need to be specified. Following [91], we use a prior trained model and set $\underline{\mu}_{io}$ to the corresponding mixture component mean in this seed model. That is, the prior knowledge before taking into account any adaptation data from the new person gives the best guess for the component mean in the new model as the component mean in the seed model. Instead of explicitly specifying Σ_{io}, we note that if we assume diagonal covariances throughout, the element-wise operation (for D-dimensional features) in (3.15) is,

$$\begin{aligned}\hat{\mu}_{id} &= \frac{\sigma_{iod}^2 \sum_{t=1}^{T} \gamma_t(i)^k x_{td} + \sigma_{id}^2 \, \mu_{iod}}{\sigma_{iod}^2 \sum_{t=1}^{T} \gamma_t(i)^k + \sigma_{id}^2} \\ &= \frac{\sum_{t=1}^{T} \gamma_t(i)^k x_{td} + \tau_{id} \, \mu_{iod}}{\sum_{t=1}^{T} \gamma_t(i)^k + \tau_{id}} \;, \text{ for } d = 1, \cdots, D \qquad (3.18)\end{aligned}$$

where $\hat{\mu_{id}}$, x_{td} and μ_{iod} are the d-th elements of the vectors, $\hat{\underline{\mu}}_i$, $\underline{x}_t$ and $\underline{\mu}_{io}$, respectively; σ_{iod}^2 and σ_{id}^2 are the d-th diagonal elements of the matrices, Σ_{io} and Σ_i, respectively; and $\tau_{id} = \sigma_{id}^2/\sigma_{iod}^2$. For simplicity if we assume τ_{id} to be identical for all the elements, i.e. $\tau_{id} = \tau_i$, for $d = 1, \cdots, D$, we obtain,

$$\underline{\mu}_i^{k+1} = \frac{\sum_{t=1}^{T} \gamma_t(i)^k \underline{x}_t + \tau_i \underline{\mu}_{i0}}{\sum_{t=1}^{T} \gamma_t(i)^k + \tau_i} \qquad (3.19)$$

τ_i can be viewed as the uncertainty regarding our prior guess for the mixture component mean, as measured by the amount of scatter in Σ_{io}, relative to Σ_i. A large value for τ_i implies that our prior certainty is strong, and the prior density is sharply peaked around $\underline{\mu}_{io}$, the component mean in the seed model [91]. A ratio of either the trace or determinant of Σ_i and Σ_{io} can be used as a scalar measure of our relative prior certainty. In practice, for simplicity, τ_i is constrained to be identical for all mixture components of a model, i.e. $\tau_i = \tau$, for all i.

For gesture components MShape and MSpeed, the models used in each of the components are HMMs with single Gaussian observation densities for each state, i.

Following [91], the initial state probabilities, π_i, and the state transition probabilities, a_{ij}, are assumed to be fixed and known. We further simplify the adaptation process by assuming that the covariances of the Gaussian density in each state are known. Hence only the mean of the Gaussian density in each state is modified by adaptation data. MAP estimation of the means, iteratively refined using the EM algorithm, proceeds in a similar fashion to that for Gaussian mixtures as described above. The re-estimation of the mean of the Gaussian density in state i, at the $(k+1)$-th iteration is calculated with equation (3.19). Here $\underline{\mu}_{io}$ is the Gaussian mean in the corresponding state i of a prior trained HMM seed model; $\underline{x}_t$ is the adaptation data available from the new signer, and this is summed over the length of the observation sequence of the known category that we are training for, and over multiple sequences if available; $\gamma_t(i)^k$ is the probability of state i, given the observation sequence and the model parameters after the k-th iteration, which can be calculated from the forward and backward variables, $\alpha_t(i)^k$ and $\beta_t(i)^k$ [131]; and τ_i can be viewed as the uncertainty regarding our prior guess for the state mean, relative to the covariance of the Gaussian density. For simplicity, τ_i is constrained to be identical for all the states of a HMM, i.e. $\tau_i = \tau$, for all i.

3.2.2 Adaptation of Bayesian network S1

Section 3.1.4 described how the parameters, $\boldsymbol{\theta}$, of the BN *S1* used in Step 3 of the block diagram (Figure 3.1), are learned using ML estimation. However, to obtain a robust estimate when using a limited amount of adaptation data, D, from a new test subject, we again utilize prior knowledge about the parameters in the form of a prior distribution for $\boldsymbol{\theta}$, to determine the posterior distribution $p(\boldsymbol{\theta}|D)$. As mentioned in Section 3.1.4, for a BN with all discrete nodes, Y_i $(i = 1, \ldots, n)$, the network parameters consist of the parameter vectors, $\boldsymbol{\theta}_{ij}$, of distinct multinomial distributions (for each Y_i, and for each configuration, $j = 1, \ldots, q_i$, of its parents $\mathbf{Pa}_{Y_i}$). These parameter vectors which are assumed to be mutually independent remain independent given the adaptation data D,

$$p(\boldsymbol{\theta}|D) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} p(\boldsymbol{\theta}_{ij}|D) \tag{3.20}$$

The posterior distributions for $\boldsymbol{\theta}_{ij}$ can therefore be determined independently (for each node Y_i and each configuration j of its parents $\mathbf{Pa}_{Y_i}$) [59].

A suitable prior distribution for $\boldsymbol{\theta}_{ij} = \{\theta_{ij1}, \ldots, \theta_{ijr_i}\}$ is defined as a Dirichlet distribution, $p(\boldsymbol{\theta}_{ij}) = \mathrm{Dir}(\boldsymbol{\theta}_{ij}|\alpha_{ij1}, \ldots, \alpha_{ijr_i})$, which is a conjugate prior for multinomial sampling. The observation of adaptation data, $\mathbf{x}$, from the new test subject, converts this to a posterior density which is again a Dirichlet distribution, $p(\boldsymbol{\theta}_{ij}|D) = \mathrm{Dir}(\theta_{ij}|\alpha_{ij1} + N_{ij1}, \ldots, \alpha_{ijr_i} + N_{ijr_i})$ [59]. Here, as in Section 3.1.4, N_{ijk} is the number of times $Y_i = k$ and $\mathbf{Pa}_{Y_i} = j$ occur in the observation data set. In the MAP estimation of Section 3.2.1, the parameter values which maximized the posterior distribution were taken to be the final adapted parameters. In this case, we take the expectation of the parameters. The expectation of θ_{ijk}, with respect to the posterior distribution is [59],

$$\begin{aligned}\tilde{\theta}_{ijk} &= <\theta_{ijk}>_{p(\boldsymbol{\theta}_{ij}|D)} \\ &= \int_{\boldsymbol{\theta}_{ij}} \theta_{ijk}\mathrm{Dir}(\boldsymbol{\theta}_{ij}|\alpha_{ij1}+N_{ij1},\ldots,\alpha_{ijr_i}+N_{ijr_i})\,d\boldsymbol{\theta}_{ij} \\ &= \frac{\alpha_{ijk}+N_{ijk}}{\sum_{k=1}^{r_i}\{\alpha_{ijk}+N_{ijk}\}} \qquad (3.21)\end{aligned}$$

Comparing (3.21) to (3.4), it can be seen that the parameters of the prior distribution, α_{ijk}, for $k = 1, \ldots, r_i$, act as additional counts for the number of times $Y_i = k$ and $\mathbf{Pa}_{Y_i} = j$ have occurred. Indeed, one of the ways of specifying the values of these parameters is to determine the number of counts that is equivalent to our prior knowledge about the process modeled by the BN [59].

3.3 Experimental Results

The dataset was generated from 8 persons (test subjects A to H), each of whom performed about 10 repetitions of each of the 20 distinct complete gesture meanings, giving a total of 1855 gesture sequences.

3.3.1 Experiment 1 - Signer-Dependent System

For the signer-dependent[2] systems, we trained the six component-level classifiers (as described in Section 3.1.3) on roughly $\frac{2}{3}$ of the gesture sequences obtained

[2] "Signer" is used here as a convenient term although the gesture set used does not contain actual signs.

from one test subject (e.g. A). The trained (signer-dependent) classifiers then output the component category labels as discrete values to network *S1*'s observation nodes, L^{HS}, L^{HE}, L^{MSh}, L^{MO}, L^{MSz} and L^{MSp}. The numerical values of the local conditional probabilities of the nodes in *S1* were learned with ML estimation (3.4). In the testing procedure, we used the trained component-level classifiers to obtain the observation node categories for the remaining $\frac{1}{3}$ of the gesture sequences, and presented these as evidence to the *S1* network for inferring the most probable values for the query nodes BasicMeaning, Intensity, Distance, Rate, and Continuance. A test sequence was considered to be recognized correctly only if all the query node values were inferred correctly. The above procedure was repeated individually on all 8 test subjects. Accuracy results ranged from 88.2% to 95.7%, with an average accuracy of 92.2% (see Table 3.2). Since the basic meaning class labels are grouped in a separate node from the inflections, we also performed "partial" recognition of the basic gesture meaning only, which yielded an average accuracy of 98.5%.

For comparison, we implemented a direct multiplication of the probability scores of component features (in the manner of [134]) and obtained a much lower average gesture recognition accuracy of 69.5%.

Network *S1* is able to take advantage of redundancies in the information from different components to disambiguate uncertainties in the component classification results. The network learns to characterize the error performance of the classifiers

Table 3.2: Gestures recognition accuracy results on test data for signer-dependent system of Experiment 1.

Test subjects	Accuracy results
A	92.5%
B	91.9%
C	92.1%
D	88.2%
E	90.1%
F	92.9%
G	93.8%
H	95.7%
Average	92.2%

and also improves the overall accuracy. As a result, though the worst performing (HOrienS component) and the best performing component classifier (MSpeed component) and had average (over the 8 test subjects) accuracies of 68.3% and 91.6% respectively, the overall gesture recognition accuracy was between 88.2% and 95.7%.

3.3.2 Experiment 2 - Multiple Signer System

Our first attempt at building a system for recognizing gestures from multiple signers used the same methodology as in Experiment 1, the sole difference being that data from 4 persons (A,B,C, and D) was pooled together for training and testing. This yielded an accuracy of 78.7% on the test data. The accuracy dropped as compared to Experiment 1 because when data from multiple persons is used, there is an increase in the variance of the class-conditional densities in the component-level classifiers (due to different styles in gesturing and body size) and these densities start to overlap (Figure 3.7).

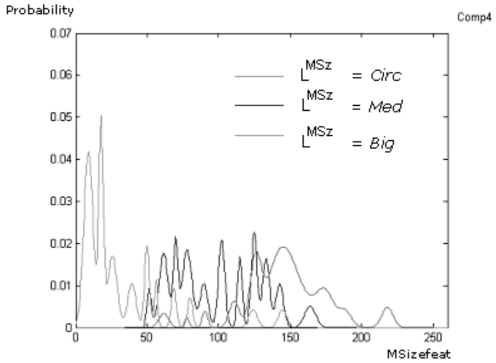


Figure 3.7: Class-conditional density functions $p(\mathrm{x}^{MSz} \mid \mathrm{L}^{MSz})$ estimated by pooling together data from 4 test subjects, A, B, C and D. There is significant overlap among the densities.

In an effort to improve accuracy, we first noted that our approach for classification at the component-level is analogous to inferencing with a generative classification model (e.g. *S2* in Figure 3.8(a), [70]) where the parent node values are discrete class labels, and the child node values are the continuous-valued feature vectors of the component. For example in *S2*, which classifies categories of the MSize component, the probability density functions for the x^{MSz} node are $p(\mathrm{x}^{MSz} \mid \mathrm{L}^{MSz})$, for $\mathrm{L}^{MSz} = Circ, Med, Big$. If MSize categories are assumed to have equal prior probabilities, inferring the most probable value of L^{MSz}, $\underset{\mathrm{L}^{MSz}}{\operatorname{argmax}} P(\mathrm{L}^{MSz} \mid \mathrm{x}^{MSz})$ (as is performed in generative classification models), is equivalent to finding the class with the highest likelihood, $\underset{\mathrm{L}^{MSz}}{\operatorname{argmax}} p(\mathrm{x}^{MSz} \mid \mathrm{L}^{MSz})$ (as performed in our approach to component-level classification). Next, to account for the person specific variations we added a PersonId node to network *S2* to obtain *S3* (Figure 3.8(b)). Hence, instead of pooling together data from multiple test subjects to estimate class-conditional densities $p(\mathrm{x}^{MSz} \mid \mathrm{L}^{MSz})$, we estimated signer-specific class conditional densities, $p(\mathrm{x}^{MSz} \mid \mathrm{L}^{MSz}, \mathrm{PersonId})$. For example in *S3*, $p(\mathrm{x}^{MSz} \mid \mathrm{L}^{MSz}, \mathrm{PersonId} = A)$, for $\mathrm{L}^{MSz} = Circ, Med, Big$, is exactly the class-conditional density of the component-level classifier for MSize trained on data from subject A in Experiment 1 (top left plot in Figure 3.9). The other plots in Figure 3.9, show $p(\mathrm{x}^{MSz} \mid \mathrm{L}^{MSz}, \mathrm{PersonId} = B)$, $p(\mathrm{x}^{MSz} \mid \mathrm{L}^{MSz}, \mathrm{PersonId} = C)$, and $p(\mathrm{x}^{MSz} \mid \mathrm{L}^{MSz}, \mathrm{PersonId} = D)$ for network *S3*.

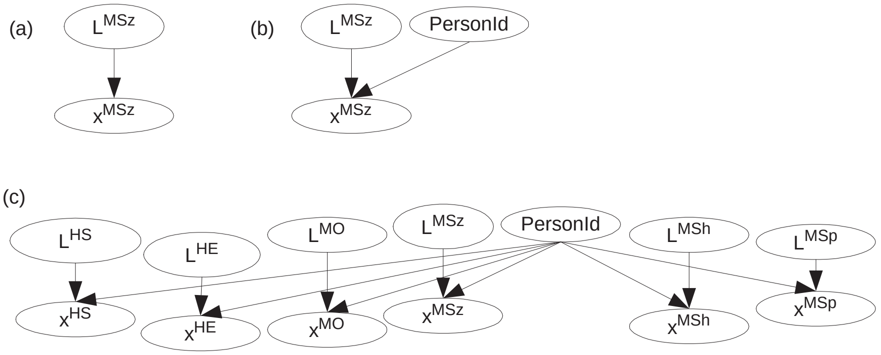


Figure 3.8: (a)*S2* for inferring L^{MSz} value. (b)*S3* which can additionally infer PersonId value. (c) *S4*, signer-indexed component-level classifier for multiple signer system.

We made the PersonId node common to all the components (since for a given gesture sequence, the same person would have produced the features in all the components) and obtained the structure *S4* shown in Figure 3.8(c). Just as $p(\mathrm{x}^{MSz} \mid \mathrm{L}^{MSz}, \mathrm{PersonId})$, are exactly the class-conditional densities of the component-level

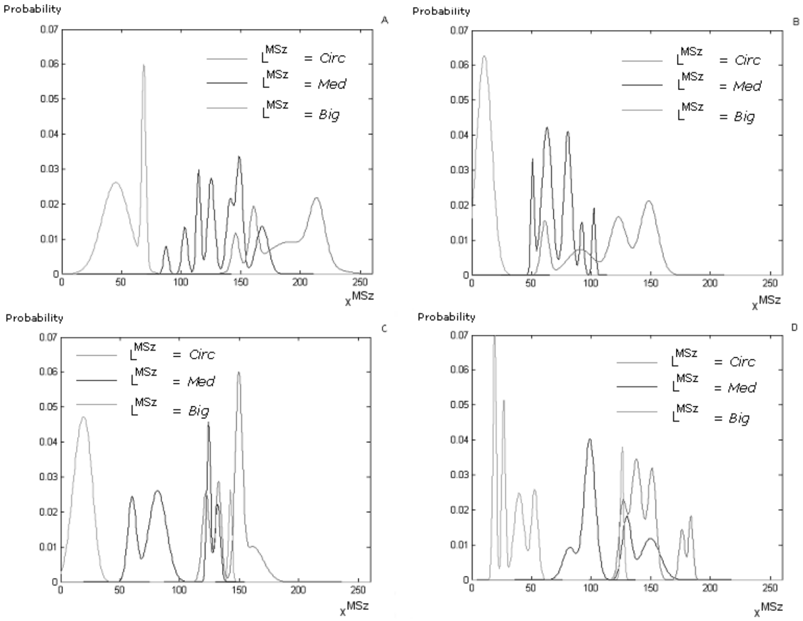


Figure 3.9: Signer-specific class-conditional density functions, $p(\mathrm{x}^{MSz}|\mathrm{L}^{MSz}, \mathrm{PersonId} = A)$, $p(\mathrm{x}^{MSz}|\mathrm{L}^{MSz}, \mathrm{PersonId} = B)$, $p(\mathrm{x}^{MSz}|\mathrm{L}^{MSz}, \mathrm{PersonId} = C)$, $p(\mathrm{x}^{MSz}|\mathrm{L}^{MSz}, \mathrm{PersonId} = D)$, in network *S3*.

classifier for MSize trained on data from signer-specific data (as described above), so also the local conditional probabilities for the other feature nodes in network *S4* are the class-conditional densities of the appropriate component-level classifiers trained on data from signer-specific data in Experiment 1. In addition, the categories in each of the L^{HS}, L^{HE}, L^{MSh}, L^{MO}, L^{MSz} and L^{MSp} and PersonId nodes are specified as equiprobable. Hence no additional training is required to obtain the parameters of network *S4*. We then followed the same $\frac{2}{3}$:$\frac{1}{3}$ data split for obtaining the training and test sets and trained the network *S1*. Following this, gesture accuracy results on the test set were obtained from networks *S4* and *S1*,

Table 3.3: Accuracy results of multiple signer system on test data in Experiment 2. Person identity is inferred from the signer-indexed component-classifier *S4*. Gesture is recognized by using the trained *S1* network to infer values of query nodes from the classification results of *S4*.

Test subjects	Gesture recognition	Person identity recognition
B,C,D (3)	84.4%	85.4%
A,B,C,D (4)	84.9%	81.6%
E,F,G,H (4)	86.9%	78.5%
A to H (8)	85.0%	61.2%

and are given in Table 3.3. These results show that this method for multiple signers generalizes well as the number of test subjects increased from 3 to 8. For test subjects A, B, C, and D, the accuracy improved from 78.7%, when the data was simply pooled together, to 84.9% when the PersonId node was used to maintain signer-specific class conditional densities. This is a 26.8% reduction in error rate.

As using the signer's identity led to improved gesture recognition results, we also investigated the extent to which a person could be recognized by observing his gestures (in our multiple signer system). For this, the identity of the test subject was inferred from the PersonId node in network *S4*. Person identification accuracy results are shown in Table 3.3. It is seen that when the system handles 4 signers, the signer identity can be recognized with a fairly high accuracy of about 80%. However, this drops to 61.2% when the system handles all 8 signers. Person identification is not critical to the gesture recognition results but is an added feature of our multiple signer system.

3.3.3 Experiment 3 - Adaptation to New Signer

When gesture sequences from subject A were tested on the multiple signer system trained on 3 other persons (B, C, and D) in Experiment 2, a gesture recognition accuracy of only 52.6% was obtained. This is not an unexpected result as Experiment 2 suggests that there are significant inter-person differences, so that recognizing gestures from a new person without any training data from that person is difficult. As is well known, in speaker adaptation, the goal is to have a system whose performance on the new speaker approaches that of a speaker-dependent system but with much less speaker-specific training data than is required for a full speaker-dependent system [166]. Similarly, we implemented the adaptation scheme discussed in Section 3.2 in an attempt to improve gesture recognition accuracy for test subject A close to that of the signer-dependent system for A (Experiment 1 - 92.5%) while using only one set of the 20 distinct gestures from A. Our scheme separately adapts (ii) the gesture component-level classifiers and, (ii) the *S1* network, both initially trained on data from subjects B, C, and D.

For each gesture component, we adapted a set of models, using the features appropriate for that gesture component, from the data of new test subject A. Each model was adapted by iteratively refining the model parameters using the EM algorithm with MAP criterion (3.7). Only the mixture component means of the Gaussian mixtures (for gesture components with static input features) and the means of Gaussian state densities (for gesture components with time sequence data as input features and modeled with HMMs) were learned during the training/adaptation by EM. The other model parameters (for example, state density covariances, state initial and transition probabilities of a HMM model), were taken unmodified from a previously trained seed model. This is a tradeoff between accuracy and computational simplicity. However, as mentioned in Section 3.2.1, simplying the MAP adaptation scheme to only adapting the means of Gaussian state densities is reasonable start because it has been found in speech recognition that the most important speaker specific effect is related to the Gaussian means of state observation densities, rather than the state density covariances, state initial and transition probabilities [166]. The component means of the Gaussian mixtures and the means of Gaussian state densities were re-estimated in each EM iteration with equation (3.19) where prior mean (for example, $\underline{\mu}_{io}$ for state i of a HMM model) was initialized to the corresponding mean of the seed model. The value of τ, in the re-estimation equation was empirically determined by experimenting with different values of τ and testing the adapted set of models on the adaptation data from subject A. In each gesture component, the value of τ that gave the best classification performance on the adaptation data was used. The final values ranged between 1 and 2.

To obtain the seed models, we first found in each gesture component, the training subjects' (B, C and D) signer-dependent classifier that gave the best classification results for the adaptation data. The appropriate set of models in each component was then used as seed models for the adaptation process. This differs from the implementation in speech recognition systems which generally use HMMs trained on multiple speakers (termed as speaker-independent models) as seed models. Our method of selecting seed models again reflects the implication from Experiment 2 results that there are significant inter-person differences in gesturing. Hence it may be more effective to tune the best performing signer-dependent classifer with the limited amount of adaptation data available rather than adapt the multiple signer gesture component-level classifier (network *S4*), which contains a greater number of model parameters.

After the gesture component-level classifiers were adapted, their classification outputs for the adaptation data were used as the discrete values of the observation nodes in the *S1* network, which together with the known values of query nodes, BasicMeaning, Intensity, Distance, Rate and Continuance, constituted the observed number of counts for node values in *S1.* These are the N_{ijk} terms in the parameter adaptation equation (3.21). As mentioned, the terms, α_{ijk} for $k = 1, \ldots, r_i$ act as prior counts for node values. We applied equation (3.21) by taking as prior knowledge the training data from subjects B, C and D that was used to train network *S1* in the multiple signer system in Experiment 2. So the term α_{ijk} is the number of times $Y_i = y_i^k$ and $\mathbf{Pa}_{Y_i} = \mathbf{pa}_{Y_i}^j$ occured in that training set.

The gesture recognition accuracy for test subject A using the adapted component classifers and adapted *S1* network increased substantially to 88.5%. Although still short of the 92.5% accuracy of the signer-dependent system for subject A, this is a significant improvement over the 52.6% recognition rate from the unadapted system trained on subjects B, C and D.

3.4 Summary

This chapter presented experiments with a simulated vocabulary of 6 lexical signs and 5 possible grammatical inflections which modify movement both spatially and temporally. Isolated gestures were used and data capture was by video camera. Although the vocabulary here is an artificial one, it has been designed to have a structure similar to SL, and thus represents a proof of concept of ideas that can be applied to recognition of continuous signing. The extension of these ideas to continuous signing of ASL sentences will be shown in the next chapter. The main ideas are as described below.

Each of the parallel and simultaneous components in gestures, has a limited number of categories/classes. Separate component-level classifiers can be trained to recognize the classes from independent feature sets. This approach simplifies classifier design – in the experiments there were only 3 to 6 categories to distinguish in each gesture component (even though the vocabulary has 20 distinct gestures), thus requiring fewer parameters and less training data. It was possible to use static features and classifiers in some components despite the fact that gesture is an inherently time sequential process.

The dependencies between basic meaning and inflection information, and gesture components were probabilistically instead of deterministically modelled. Due to conditional independence assumptions (as embodied in the *S1* BN structure of Figure 3.6(c)), the parameters encoding the numerical values of these dependencies are estimated independently. We need never model the interaction between gesture components. At the same time, the advantage of the probabilistic approach can be seen from considering the results obtained from the recognition of individual signer gestures with a signer-dependent system in Section 3.3.1. Here we obtained average recognition accuracy of 92.2%, even though the worst-performing and best-performing component-level classifiers had accuracies of 68.3% and 91.6%, respectively. Our approach improved on component-level classifier accuracies, whereas combining the component outputs by assuming dictionary definition of signs could only yield an average gesture recognition accuracy of 69.5%.

In addition to the main points above, we also developed an additional network (*S4*) to account for differences among signers by characterizing probability densities of component features according to signer identity. This approach was found to generalize well as the number of test subjects were increased from 3 to 8. We also considered the problem of adapting the models trained on three test subjects with a small amount of data from a fourth person. The approach is novel in that we adapt both the component-level classifiers as well as the BN that combines component-level classifier outputs. The component-level classsifier were adapted using a variation of maximum a posteriori (MAP) adaptation, one of the main speaker adaptation schemes in speech recognition systems. The BN was adapted by representing the parameters of the trained system as a Dirichlet prior. A further advantage of having separate components of information with only a few categories in each is that, although there is only one set of the 20 distinct gestures available as adaptation data, there can be multiple instances available for adapting each category of classifiers. For example, 2 to 12 instances were available for adapting the HMMs in the MShape component.

Chapter 4

Recognition of continuous signing with dynamic Bayesian networks

The main ideas that were presented in the previous chapter for recognizing isolated gestures are used to develop models to recognize continuously signed ASL sentences that include inflected signs. These ideas are as follows:

- Signs/gestures can be decomposed into parallel and simultaneous components.
- Each of these components consists of a limited number of categories or values. The component-level classifiers are trained independently of each other using independent feature sets.
- It is advantageous to model sign/gesture to component dependencies probabilistically rather than deterministically.

- The sign/gesture to component dependencies can be learned from data (instead of specified using domain knowledge). We assume conditional independence such that this learning is done separately for each component.

In continuous signing, the goal is to recognize the sequence of signs in a sentence. Each of the signs in turn consists of synchronized sequences of distinct values in each sign component (see Section 2.2), and within each component, classification of the features into a distinct component value requires observing a sequence of such features. Thus the extension of the main concepts above to modelling and recognition of continuously signed ASL sentences requires a model for sequential data and synchronization (at sign boundaries) between component feature/data streams.

In the isolated gesture experiments of Chapter 3, a Bayesian network (BN) was used to model gesture to component dependencies. Each of the observation nodes of the BN, which represented gesture component values, had only one input value for each gesture even though it is inherently a time sequential process. This is because the component-level classifiers (some of which take as input a sequence of features) output one component value for each gesture. Thus the model for gesture to component dependencies did not require temporal modelling of the data and the (static) BN was an adequate model. The dynamic Bayesian network (DBN) is an extension of the BN for modelling temporal process. The next section describes

DBNs in general and the application of a simple DBN, the hidden Markov model (HMM), to speech and sign recognition. Section 4.2 describes the hierarchical hidden Markov model (H-HMM), which is a DBN structure suitable for modelling the hierarchical structure in speech. Although the sign data stream has a similar hierarchical structure, it also has a parallel and simultaneous structure and hence can be decomposed into multiple streams of component features/data. Section 4.3 examines some of the DBNs that have been used for modelling and combining multiple data streams. We then present a new DBN structure in Section 4.4 called the Multichannel Hierarchical Hidden Markov model (MH-HMM) which models both the hierarchical structure and the parallel and simultaneous component data streams in signing. In Section 4.5 we show how the MH-HMM can be applied to the modelling and recognition of sign sentences that include inflected words.

4.1 Dynamic Bayesian networks

A dynamic Bayesian network (DBN) [50, 109] can be used to represent random proccesses, i.e. random variables that evolve with time. Each of the random variables is either hidden or observed. The hidden state of the system at time t is represented in terms of a set of hidden variables, $X_{k,t}$, $k = 1, \ldots, K$. There are also multiple observation variables at time t, $Y_{l,t}$, $l = 1, \ldots, L$. The DBN is a graphical model consisting of nodes representing these variables and directed edges representing dependencies among the nodes. The edges link nodes that are within the same time slice or across two consecutive time slices. Just as in the BN, absence of edges in the graph implies conditional independence, i.e. a node is independent of its non-descendants, given its parents. This allows the joint distribution of the random variables represented by the nodes to be factored as a product of local conditional probability distributions (CPD). The key difference from a BN is that in the DBN, there is a set of random variables $\{X_{1,t}, \ldots, X_{K,t}, Y_{1,t}, \ldots, Y_{L,t}\}$ at every time slice t. The models are taken to be 1st-order Markov, so that the parents of any one variable are either from the same time slice or the previous time slice. The CPDs are taken to be time-invariant to allow modelling of arbitrary length sequences with a limited number of parameters. As a result of the 1st-order Markov and time invariance properties, we only need to define the CPD for variables in a DBN unrolled for the first two time slices [109].

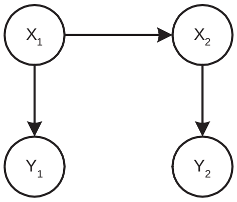


Figure 4.1: DBN representation of a HMM, unrolled for the first two time slices.

The simplest DBN is a HMM, which at time t has a single hidden variable, X_t, and a single observation, Y_t. Figure 4.1 shows the DBN representation of a HMM. For a HMM whose hidden variable X_t can take on M possible values/states, the parameters (for the CPDs) required to specify the model are [131]:

- state initial probabilities, $\pi_i = P(X_1 = i)$, for $i = 1, \dots M$.
- state transition probabilities, $a_{ij} = P(X_{t+1} = j | X_t = i)$, for $i = 1, \dots M$ and $j = 1, \dots M$.
- output probability distributions, $b_i(y_t) = P(Y_t = y_t | X_t = i)$, for $i = 1, \dots M$.

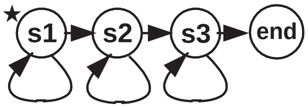


Figure 4.2: State transition diagram of an example HMM phone model with three states. Initial state probabilities are zero for all but the s1 state. Thus only the s1 state can be joined to states of the previous phone model when they are chained together in the HMM recognition model. The *end* state is not an actual state, it just identifies which state of this model (in this case only the s3 state) can be joined to states of the next phone in the recognition model (see text for explanation).

HMMs are widely used in speech recognition systems, where they are able to process speech utterances of variable lengths and implicitly segment continuous speech into individual words. Generally one HMM is trained to model each basic sound or phone in the spoken language (see Figure 4.2 for the state transition diagram of such a phone model, not to be confused with the DBN representation in Figure 4.1). All words can be decomposed into a sequence of these phone subunits which are limited in number (for example there are 42 units in English [34]), thereby enabling recognition of large vocabularies with a finite number of trained HMMs. During recognition, the trained HMM phone models are chained together into a branching tree-structured network that allows all valid word sequences –

called a recognition model [18]. Viterbi decoding is used to find the most probable state path through the HMM recognition model, thereby recovering both the word boundaries and sequence [173]. This idea has also been used for recognition of continuous signs ([47, 103, 143]); some of these works define sequential subunits for the same purpose as phone subunits for speech recognition, i.e. reducing training data requirements and scaling to large vocabularies ([12, 13, 157, 161, 174]).

However, the HMM has the disadvantage of being a flat model where all the information about the state of the system is contained in a single, unfactorised state variable. In speech recognition for example, this state variable identifies the word, the phone and the state within the phone model. As mentioned in [110], there are disadvantages to using a flat structure such as the HMM to model the essentially hierarchical structure inherent in speech:

1. Modularity in the parameters is lost. For example the dependencies between word to phone, and from phone to subphone (HMM state) are combined in a complex way in the flat HMM structure. In HMM-based speech recognition systems, the word model (i.e. the decomposition of a word into a phone sequence) is most often determined according to a pronunciation dictionary. If a particular word has only one possible pronunciation, the word to phone sequence dependency would be deterministic; if multiple pronunciations are

possible the dependency is probabilistic. Regardless, the dependency is usually assumed to be fixed and in the flat HMM structure there is no modular way of learning or adjusting the probabilistic dependencies between word to phone.

Learning the probabilistic dependencies between word to phone may be desirable in speech recognition because dictionary definitions, no matter how comprehensive, cannot account for all the variations in pronunciations due to accent, regional differences or personal styles. This is however especially pertinent in the case of SL recognition because there is no commonly agreed upon definition of sign subunits and the equivalent of the pronunciation dictionary in speech is not available.

2. There is implicit sharing of phone model parameters without a clear representation. In speech, multiple words often share the same phone. However a given phone which appears in different words would generally be followed in sequence by a different phone. So it is necessary to identify not only the current phone but also the current word as well. Thus different instantiations of each phone exist corresponding to different word contexts, making the overall HMM recognition model very large [18]. At the same time, the CPD parameters for the different phone instantiations would need to be tied together – this is not represented in the HMM.

The next section describes a model that is better suited for modelling the hierarchical organization in both speech and SL manual signs, including any probabilistic relationship between a word and its associated phone sequence.

4.2 Hierarchical hidden Markov model (H-HMM)

Speech has a natural hierarchical structure where phones combine sequentially to form words, and combinations of words form sentences. Each phone is considered as a quasi-stationary process consisting of a sequence of steady-state periods, and hence, HMM-based speech recognition systems model a phone as consisting of a sequence of subphones or HMM states. Each level (sentence, word, phone, subphone) has a different time scale, and in fact the state transition time at any particular level depends on the time taken to finish a state sequence at a lower level. For example, the next word in a sentence can start only when the phone sequence of the current word has ended. Similarly, within this phone sequence, the next phone can start only when the subphone or HMM state sequence of the current phone has ended.

Hierarchical hidden Markov models (H-HMM) [44, 110] have been proposed as a suitable DBN structure for modelling domains with hierarchical processes that evolve at multiple time scales. In applications such as human activity recognition [90, 118], event and scene recognition in video sequences [99, 169], and grammatical relations recognition in text sentences [141], H-HMMs have been found to give better results than baseline HMM approaches. There are two kinds of states in a H-HMM, abstract states and production states. An abstract state does not emit any observations but calls a lower-level state, usually starting a state sequence at the lower-level. Each of the states at this level may in turn also be an abstract state which calls another lower-level state. At the lowest level are production states which emit observations. A lower-level state sequence must finish before it returns control to the higher-level state that called it.

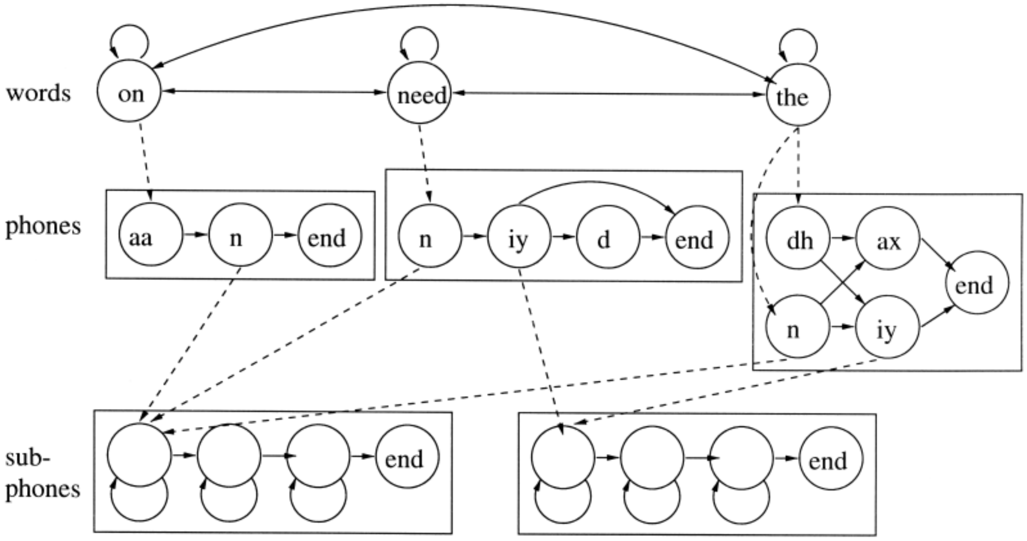


Figure 4.3: State transition diagram of an example H-HMM for a speech recognition system that can recognize three words. Phone models (represented by surrounding boxes at the 3rd level) are shared by different words – thus multiple dotted-line arrows point to the starting state of the same phone model (only two phone models are shown to avoid clutter). The subphones are equivalent to HMM states and are the only states that emit observations. The *end* states are not actual states, they just identify which states of a particular model can be the last state in the state sequence for that model (from [111], adapted from [73]).

To model speech using a H-HMM, we can represent the word and phone values as abstract states and subphones as the lowest level production states. Figure 4.3 shows the state transition diagram for an example H-HMM modelling three words. Here solid-line arrows represent horizontal transitions within the same level, while dotted-line arrows represent vertical transitions, i.e. calls to a lower-level state. Consider an example of generating a sentence from this H-HMM. Say the first word in the sentence is “on”. This triggers the phone sequence associated with this word, i.e. the word model for “on”. The first phone in this sequence is “aa”. The call to “aa” in turn triggers the subphone sequence of this phone, the phone model for “aa”. Each of the subphones[1] in sequence emits an acoustic vector. When the subphone sequence for “aa” reaches its end, it returns control to the phone-level which then goes on to the next phone “n”, triggering the subphone sequence of the phone model for “n”. When this new subphone sequence ends, control again returns to the phone-level. At this point, the phone sequence of the “on” word model has reached its end. Control thus returns to the word-level where the next word can either be the same word (“on”), or a different word, “need” or “the”. Once the next word is chosen, the phone sequence of that word model is triggered and the same process as above ensues. The word sequence reaches its end at the end of the sentence. (The *end* state at the word-level is not shown in Figure 4.3).

[1] In the rest of the chapter, the word *subphone* is used exclusively to refer to the HMM states of a phone model. The word *state* will refer to either the abstract and production states at different levels of a H-HMM or the state of the entire DBN. The meaning will be clear by context.

The *phone model* is defined by the decomposition of a phone into its associated subphone sequence and the output probability distributions for the subphones. The associated subphone sequence for a phone is defined by the subphone initial and transition probabilities (equivalent to the HMM state initial and transition probabilities of Section 4.1), and the subphone ending probabilities (the probability of each subphone being the last in the sequence). We refer to this as the state initial, transition and ending probabilities at the subphone-level which is represented in Figure 4.3 by the state transitions within the surrounding boxes at the 3rd level. There is one set of such probabilities for each phone model. The *word model* is defined as the decomposition of a word into its associated phone sequence, which is defined by the state initial, transition and ending probabilities at the phone-level. This is represented by state transitions within the surrounding boxes at the 2nd level. There is one set of such probabilities for each word model. Lastly the *sentence model* is defined as the set of valid word sequences that can be constructed from the H-HMM. It is defined by the state initial, transition and ending probabilities at the word-level. For example the sentence model in Figure 4.3 shows that any of the three words can start a sentence, and each of the words can be followed by any of the other two words as well as by itself (the ending probabilities are not represented in the figure).

The H-HMM for speech recognition can be represented by a DBN such as in Figure 4.4 which shows the DBN unrolled for the first two time slices [111].

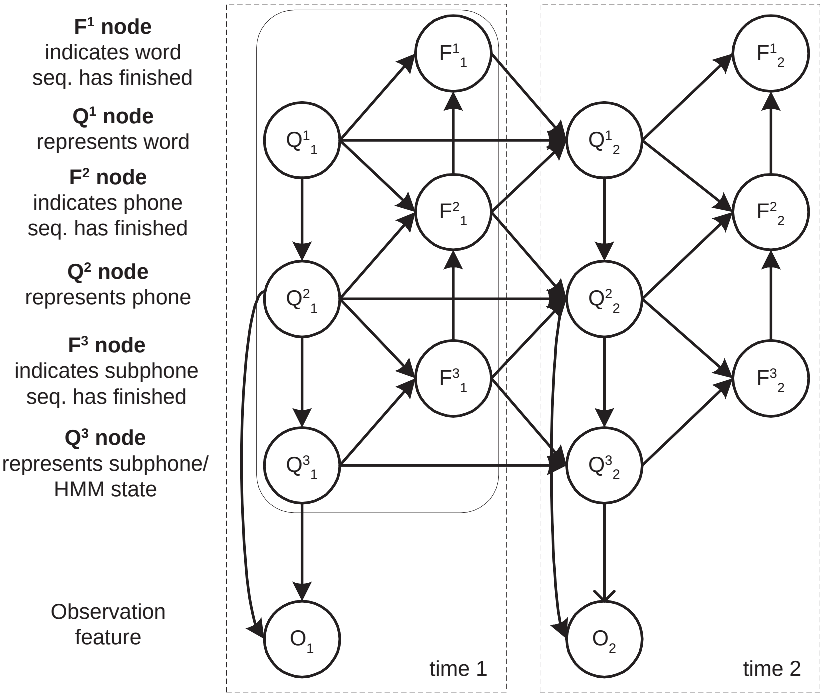

F¹ node indicates word seq. has finished
Q¹ node represents word
F² node indicates phone seq. has finished
Q² node represents phone
F³ node indicates subphone seq. has finished
Q³ node represents subphone/ HMM state
Observation feature
F¹₁
Q¹₁
F²₁
Q²₁
F³₁
Q³₁
O₁
time 1
F¹₂
Q¹₂
F²₂
Q²₂
F³₂
Q³₂
O₂
time 2

Figure 4.4: H-HMM for speech recognition (from [111]). Dotted lines enclose nodes of the same time slice.

The word, phone and subphone at time t are represented by Q_t^1, Q_t^2, and Q_t^3 respectively, collectively called the Q nodes. The value of the Q_t^d node, is the state at level d and time t. In a sense, each level implements a set of models, with the exact model that is currently active dependent on the value of the higher-level state. For example, the word value at time t, q_t^1, determines the current word model that is active at the phone-level, i.e. the phone sequence associated with the word value is implemented. Similarly, the phone value at time t, q_t^2, determines the current phone model that is active at the subphone-level, i.e. the subphone sequence associated with the phone value is implemented. Hence, in general, Q_t^d is necessarily a parent of Q_t^{d+1}.

The F_t^d nodes are binary indicator variables with a value of 1 ("on") if the state sequence at level d and time t has finished [110]. Otherwise, it has a value of 0 ("off"). As an example, consider F_t^{d+1}. There are two possible situations at time t:

- The current state sequence at level $d{+}1$ has finished, indicated by the variable F_t^{d+1} being "on". Control should then return to the higher-level, d, which can now change state, i.e. Q_{t+1}^d can be a different value from Q_t^d. This then triggers a new state sequence to start at level $d+1$ and time $t+1$. This new state sequence is associated with the new d-level state, i.e. q_{t+1}^d.
- The current state sequence at level $d+1$ has not finished, the variable F_t^{d+1} is "off", and the value of Q_{t+1}^d is forced to remain in the same state as in the previous time slice, i.e. $q_{t+1}^d = q_t^d$, and a new state sequence is *not* triggered at level $d+1$ and time $t+1$.

Thus F_t^{d+1} is both a parent of Q_{t+1}^d (to indicate when its value can be different from the value of Q_t^d, i.e. when level d can change state), and a parent of Q_{t+1}^{d+1} (to indicate when its value should be drawn from the initial state probabilities of the model associated with q_{t+1}^d, i.e. when a new state sequence at level $d+1$ should be started). Lastly, F_t^{d+1} node is a parent of F_t^d node to enforce the requirement that a higher-level sequence cannot finish when the lower-level sequence has not. Collectively, the indicator nodes enforce the different time scales at each level and only allow a higher level to change state when the lower-level sequence has finished.

We now consider how H-HMMs allow modularity in the parameters and sharing of phone models by multiple words.

4.2.1 Modularity in parameters

As mentioned, the sentence model is the set of possible word sequences and is defined by the state initial, transition and ending probabilities at the word-level. In the DBN of Figure 4.3, these probabilities are encoded in the parameters of the CPDs for nodes Q_t^1 and F_t^1. The word model for a particular word is the phone-level state initial, transition and ending probabilities associated with that word. These are encoded in the CPD parameters for nodes Q_t^2 and F_t^2. Both nodes have as one of their parents, the Q_t^1 node, thus the word value determines which set of phone-level state probabilities is active. The phone model for a particular phone has an associated subphone sequence defined by the subphone-level state initial, transition and ending probabilities. These are encoded in the CPD parameters for nodes Q_t^3 and F_t^3. Both nodes have as one of their parents, the Q_t^2 node, thus the phone value determines which set of subphone-level state probabilities is active. The phone model also includes the output probability distributions for the subphones associated with it. This distribution is defined by the CPD of the observation feature O_t. Notice that both the phone node (Q_t^2) and the subphone node (Q_t^3) are parents of O_t. The output probability distribution is determined by both the phone value as well as the subphone value since for example, the first state of two different phone models would not have the same output probability distribution.

Thus we see that the probabilities defining the sentence, word and phone models are distinct and easily extracted from the node CPDs of the DBN. They are not lumped together into the state initial and transition probabilities for a single variable that represents the entire state of the system, as occurs in the HMM. Conceptually, the system's state has been factorised into the random variables enclosed by the rounded-rectangle in Figure 4.4.

4.2.2 Sharing phone models

Multiple words sharing the same phone is easily represented in the H-HMM state transitions without needing to create multiple copies of the same phone (see Figure 4.3). The word node (Q_t^1) is not a parent of the subphone-level Q and F nodes (Q_t^3 and F_t^3), thus the subphone-level state initial, transition and ending probabilities are dependent only on the phone node (Q_t^2), and multiple words in which a phone occurs share the same model of the phone. At the same time, control over the "flow" of phone sequences is maintained by the phone-level state initial, transition and ending probabilities which do depend on the word value since the word node (Q_t^1) is a parent of the phone-level Q and F nodes (Q_t^2 and F_t^2).

The H-HMM models the hierarchical structure in speech and would similarly be able to model the hierarchical structure in SL sentences. However SL manual sign sequences not only exhibit hierarchical structure, they also consist of multiple data streams, corresponding to each sign component. Hence, in the next section we review some models for combining and modelling multiple data streams.

4.3 Related work on combining multiple data streams

Various statistical models have been proposed to handle problems where multiple observation streams correspond to the same sequence of events. The information streams that are combined and their corresponding application domains include: different acoustic features for speech recognition [176], acoustic phone features and pitch features for recognition of Mandarin tonal phones [92], clean speech and noise for speech recognition [155], different frequency bands for speech recognition [21, 32, 60, 52, 106, 120], audio and visual features for speech recognition [15, 28, 39, 53, 54, 72, 98, 105, 114, 115, 117, 127, 151, 177], features of different sign components for recognizing manual signing in SL [157], features from two hands or individuals for gesture/action recognition [26], data from video, audio and computer interactions for office activity recognition [121], audio and visual features and features from individual participants for recognition of group actions in meetings [102, 175], features from multiple individuals for human interaction recognition [122], audio and visual detector outputs for speaker detection [48], different facial features for facial expression recognition [170], different body parts for action recognition [123].

In the following we will examine the statistical models used in the above applications in terms of how they deal with the issues related to modelling and combining multiple data streams: asynchronicity between data streams, hierarchies of data and events, and the requirement to jointly train with the multiple observation streams.

4.3.1 Flat models

The HMM, as opposed to a H-HMM, is a flat model that contains all the information about the different abstract levels of a system in a single state variable. Similarly, in the flat models described below, the hierarchy of abstract levels and multiple time scales are not explicitly modelled. These models can be divided into those that do not necessarily require concurrent training with the multiple observation streams (multistream HMM, product HMM, parallel HMM) and those that do (coupled HMM, factorial HMM, B-band DBN, asynchronous HMM).

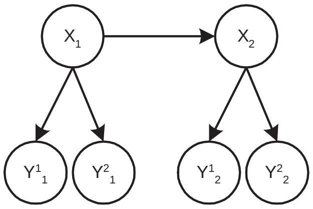
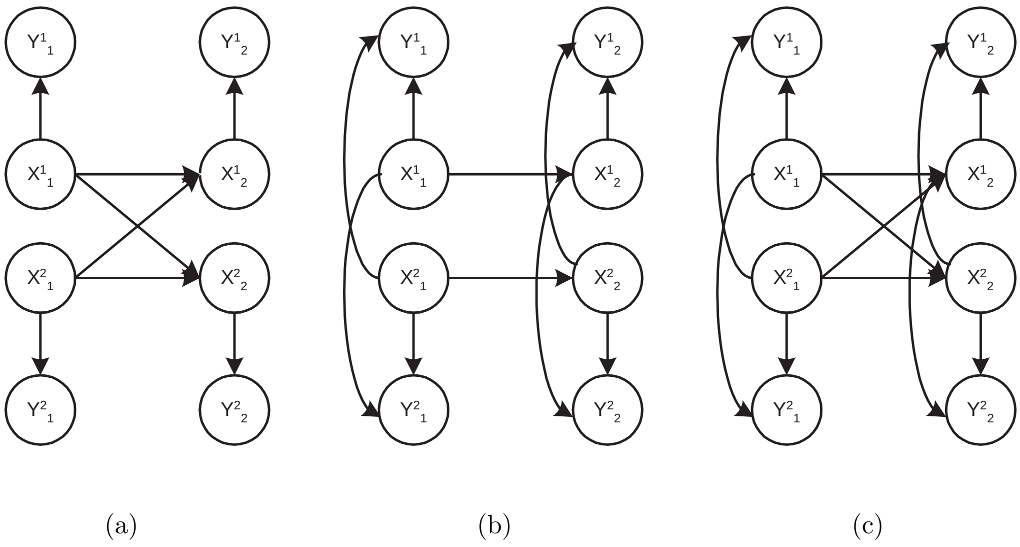


Figure 4.5: DBN representation of a multistream HMM with two observation streams, unrolled for the first two time slices. The DBN for a product HMM is identical.

In the multistream HMM [21, 98, 117], and product HMM [54, 155, 177] frameworks, individual HMMs may be trained separately for each data stream. These HMMs are then combined into a multistream HMM or a product HMM for decoding/testing. The DBN representation for these two types of models is as shown in Figure 4.5, where two observation streams with features Y_t^1 and Y_t^2, respectively, are modelled. The combined model has a single hidden state variable at any time instant. The hidden state value is the combination of the state values in each of the individually trained HMMs. In the multistream HMM, the state values that are combined from the individually trained HMMs are forced to be identical. Thus the modelling assumption is that the two different sequences are *state synchronous.* In the product HMM, the state values that are combined from each of the individually trained HMMs can be different as long as they belong to the same model. Since in speech, HMMs usually model the phone, the modelling assumption in the case of the product HMM is that the two different sequences are *phone synchronous.* During recognition, the phone models are chained together into a branching tree-structured network as with regular HMMs and the Viterbi decoding algorithm finds the most probable state path through the network. The parallel HMM [157] makes the same assumption of synchronization at model boundaries as product HMMs but does not form a combined HMM from the individual HMMs trained on separate data streams. Instead decoding is done separately in individual HMM decoding networks or recognition models and the n-best *word* or *phone synchronous* paths (with n ranging up to 20) from each network are combined to find the best combined path. This is a suboptimal solution as there is no guarantee that the best overall path (as would be found in the product HMM framework) is among the n-best paths found in each of the individual HMM decoding networks. Another modelling paradigm that does not require concurrent training with the multiple observation streams is discriminative model combination with rescoring of n-best hypotheses [117], [19], [156]. In speech recognition, the n-best hypotheses from the separate data streams that have the same word sequence are combined. n can be quite large, for example, 2000 best hypotheses were used in [117]. Within a sentence, complete asynchrony between the individual HMMs is allowed.



(a) (b) (c)

Figure 4.6: (a) Coupled HMM, (b) Factorial HMM, (c) general loosely coupled HMM (all figures adapted from [119]).

Coupled HMMs [26] and factorial HMMs [51] are examples of loosely coupled

HMMs [119]. In these models, the state of the system, X_t, at each time instant is factorized into state variables that represent the state of the process in each of the multiple data streams. For example, X_t is factorized as X_t^1 and X_t^2 for the case of two data streams. The factorized states can have various degrees and manner of coupling and interaction. In the coupled HMM, the state transitions of the individual processes are coupled (Figure 4.6(a)). In the factorial HMM, the states of the individual processes are not coupled directly but they share the same observations (Figure 4.6(b)). In the general loosely coupled HMM, the states of the individual processes are coupled and also share the same observations (Figure 4.6(c)). Due to the coupling of the state variables and/or sharing of observations, the models mentioned above must be jointly trained with the multiple data streams as observations. Other models that also require such training includes the B-band DBN [32] and the asynchronous HMM [15].

Being able to perform training separately using the different observation data sequences and then using the learned parameters in the combined model is an advantage. Training is faster since it is performed on simpler models and there is no requirement for the training data to include all possible combinations of values in each data stream. A key step that makes this possible is enforcing synchronization at some level (for example, word-level) while allowing complete asynchrony at lower levels (for example, phone and subphone levels), as opposed to modelling the asynchrony between data streams at the state-level (as in loosely coupled HMMs) or at the level of each time slice (asynchronous HMM).

4.3.2 Models with multiple levels of abstraction

Multiple levels of abstraction are useful when a high-level event can be decomposed into a sequence of sub-processes or sub-events, as for example in speech. In the layered HMMs of [175], this concept was applied to model actions in a meeting as consisting of multiple lower-level actions by each individual participant. Two layers of HMMs were used and the posterior probabilities from multiple lower-layer HMMs (which modelled each participant's actions) were concatenated as observations of the higher-layer HMM (which modelled the meeting actions). A similar model was used in [121] to model and recognize office activities.

In the layered HMM structure, the multiple levels of abstraction are not modelled concurrently. Each level takes its observations from the previous level and generates the observations for the next level. Thus the recognition and decoding in each level occurs in a decoupled manner. The work most closely related to our proposed model in the next section are the DBN models of [53], [176] and [92] which concurrently model the multiple levels of abstraction in multiple processes and data streams. In Gowdy et. al [53] an acoustic feature stream and a video data stream are modelled to perform audio-visual speech recognition. In this model, the word transition times are solely determined by the acoustic data stream, i.e. a transition to the next word occurs when the phone sequence in acoustic stream for the current word has finished, even though the phone sequence in video stream may not have finished. Thus the acoustic stream acts like a master sync. Zhang et. al [176] which models different acoustic features for speech recognition uses the same structure as in [53], thus similarly has a master sync channel. Lei et. al [92] combine acoustic phone features and pitch features in the recognition of Mandarin tonal phones. This work is different from ours in that it recognizes phone sequences and not word sequences. Also, no details were given for the CPD parameters required for the phone-level F_t node, which is crucial for synchronization between the multiple data streams.

4.4 Multichannel Hierarchical Hidden Markov Model (MH-HMM)

The analysis of SL manual sign structure presented in Section 2.2 represents signs as parallel and simultaneous sequences of values in each of the sign components. There is a limited number of “primes” or classes in each of the components, which we can consider as the equivalent of phone subunits in speech. So a sign is decomposed as a sequence of phones in each component stream. As mentioned in Section 2.2, there is no requirement for the different streams to synchronize at the phone-level or any other sub-sign “segmental” level (such as implied in Liddell’s model [95]). The only requirement is synchronization at the sign-level, i.e. for any particular sign in the sentence, the phone sequence for that sign in each component stream should start and end at the same time. In the actual physical performance of signs, it is likely that at sign boundaries, the phone values across components do not synchronize exactly at the per-frame level. However, this synchronization constraint is necessary for connecting phone variables across components to the same parent sign variable and in our view is more reasonable than allowing sign transition times of different components to be completely unconstrained, as has been implemented in [157].

We propose the Multichannel Hierarchical Hidden Markov model (MH-HMM) as a DBN suitable for simultaneously modelling both the hierarchical and the parallel structure in sign sequences. This structure is shown in Figure 4.7. The MH-HMM models a sentence as made up of a sequence of signs, and each sign as made up of parallel phone sequences, one in each sign component. Additionally, a phone in a component may be decomposed as a sequence of subphones. Most of the previous work in combining multiple data streams either modelled a flat structure for the parallel data streams, or where multiple time-scales and a hierarchical structure was considered, modelled the higher and lower-levels of the hierarchy in a decoupled manner. In contrast, the MH-HMM models multiple data streams with hierarchical structure, and different levels of the hierarchy are jointly modelled. In addition, sign-level synchronization between component streams is accomplished through the use of a sync node, S_t^2 in Figure 4.7, such that none of the components have priority in terms of synchronization. This is unlike the models proposed in

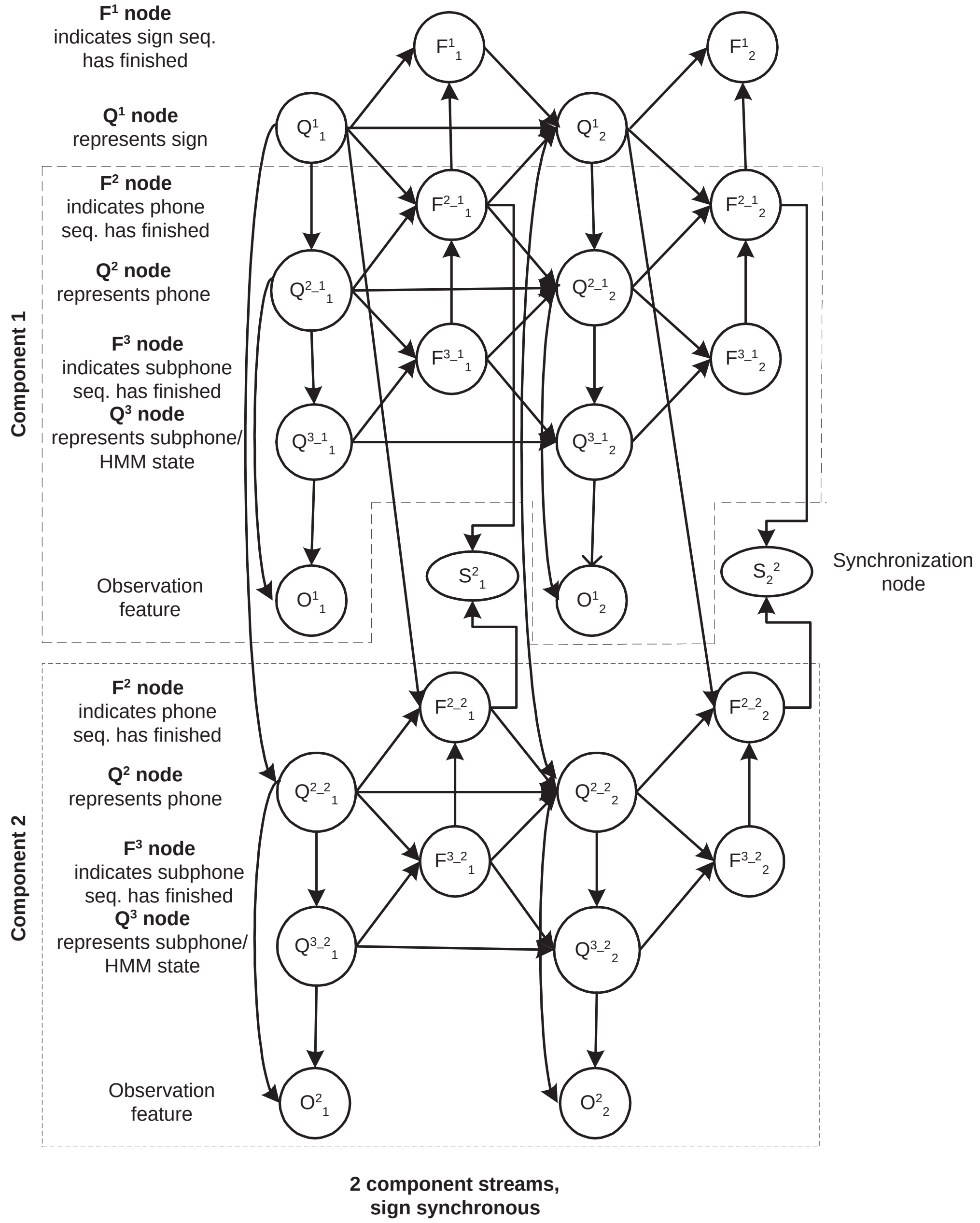


Figure 4.7: MH-HMM with synchronization between components at sign boundaries (shown for a model with two components streams, and two time slices). Dotted lines enclose component-specific nodes.

[53] and [176] where one of the data streams is the master sync. Another advantage of the MH-HMM framework is that it allows training to be performed separately on each component's observation feature stream. The training process will be covered in more detail in Section 4.4.2 and Chapter 6.

The key difference between the MH-HMM and the H-HMM is that in the MH-HMM, there is one set of sign-level nodes, Q_t^1 and F_t^1, but multiple sets of phone- and subphone-level and observation feature nodes. Figure 4.7 shows a model for two component streams where there are two sets each of $Q_t^{2\text{-}c}$,$Q_t^{3\text{-}c}$, $F_t^{2\text{-}c}$, $F_t^{3\text{-}c}$, and O_t^c nodes, with $c = 1, 2$. In general, we can expand the model to as many sets, N_c, of the above nodes as required to model multiple component data streams. In Section 4.4.1, we show how parameters for the MH-HMM can be learned by training separately with each of the component data streams. Thus for example, the MH-HMM model of Figure 4.7 is only employed during testing/decoding, with much simpler models used during training. In our approach, training complexity increases linearly with the number of component data streams that are modelled.

The phone-level nodes ($Q_t^{2\text{-}c}$ and $F_t^{2\text{-}c}$, $c = 1, \ldots, N_c$) share the same parent sign node (Q_t^1). So at any instant in time, the phone sequences in each component are associated with a common sign value. However, each component c has a separate set of phone-level nodes ($Q_t^{2\text{-}c}$ and $F_t^{2\text{-}c}$, $c = 1, \ldots, N_c$), subphone-level nodes ($Q_t^{3\text{-}c}$ and $F_t^{3\text{-}c}$, $c = 1, \ldots, N_c$) and observation feature nodes (O_t^c, $c = 1, \ldots, N_c$). So within the time period of a sign, the different component data streams can have

Table 4.1: CPD for the sign synchronization node S_t^2 in a MH-HMM modelling three components. The CPD implements the EX-NOR function.

			$P(S_t^2 \mid F_t^{2\_1}, F_t^{2\_2}, F_t^{2\_3})$	
$F_t^{2\_1}$	$F_t^{2\_2}$	$F_t^{2\_3}$	$S_t^2 = 0$	$S_t^2 = 1$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

different phone and subphone state evolution dynamics, where the phone values in one component stream may be changing faster or slower than those in another component stream. At sign boundaries however, the phone sequences for the current sign in all N_c components are required to end, and the phone sequences in all components for the movement epenthesis that links up to the following sign must start. In the MH-HMM, this is achieved by forcing $F_t^{2\_c}$ (which indicates when the phone sequence of the c-th component has ended), for $c = 1, \ldots, N_c$, to all have values of 0 or all have values of 1. We introduce a synchronization node S_t^2, as the common child of the $F_t^{2\_c}$ nodes. The CPD of S_t^2 is defined as the EX-NOR function (see Table 4.1), so that $S_t^2 = 1$ only when its parents either all have values of 1 or all have values of 0. When the MH-HMM is used for recognizing continuous signing, for example, when we input the data from a test sentence, we set $S_t^2 = 1$ in all time slices to enforce sign-level synchronization.

We also mention here that the synchronization node between components can be applied at other levels. For example, it can be made a child of all the $F_t^{3\text{-}c}$ nodes. This would enforce all component streams to transit from the current phone to the next phone at the same time. Since in our analysis in Section 2.2 we require synchronization at the sign-level only and not at the phone-level, we apply the synchronization node as noted above.

All the advantages of the H-HMM versus the flat-HMM as mentioned in Section 4.2 apply as well to the case of MH-HMM versus flat models for combining multiple data streams, i.e. modularity in parameters and sharing of phone models between different signs.

4.4.1 MH-HMM training and testing procedure

In the MH-HMM, the sentence model, i.e. the possible sign sequences, are encoded in the CPD parameters of the sign-level nodes Q_t^1 and F_t^1. This is similar to the case for the H-HMM as used in speech modelling (see Section 4.2.1). In speech modelling, the word model for a particular word is the phone-level state initial, transition and ending probabilities associated with that word. In the MH-HMM however, for a particular sign, there is not one but N_c component-specific sign models, one for each component, c. And the phone-level state initial, transition and ending probabilities for the c-th component are encoded in the CPD parameters for nodes $Q_t^{2\text{-}c}$ and $F_t^{2\text{-}c}$. For each phone in the c-th component, the phone model for that phone has an associated subphone sequence defined by the subphone-level state initial, transition and ending probabilities. These subphone-level state probabilities are specific to the component, and are encoded with the CPD parameters for nodes $Q_t^{3\text{-}c}$ and $F_t^{3\text{-}c}$. The output probability distributions for the subphones are also specific to the component and are defined by the CPD of the component's observation feature O_t^c. Thus in the MH-HMM, there is one common sentence model, while the sign and phone models are component-specific.

Our training and modelling strategy is to learn the component-specific sign and phone models by training each component's models independently of each other and with independent observation feature sets. This training is done using the (single channel) H-HMM (see Section 4.4.2). After training, the learned component-specific sign and phone models are combined in the MH-HMM by specifying the CPD parameters for the component-specific phone-level nodes ($Q_t^{2\text{-}c}$ and $F_t^{2\text{-}c}$), subphone-level nodes ($Q_t^{3\text{-}c}$ and $F_t^{3\text{-}c}$), and observation feature nodes (O_t^c), for $c = 1, \ldots, N_c$. The sentence model for a particular set of sentences can be straight-forwardly determined from knowledge of the sign sequences that appear in the sentence set. For example, the probability of a particular sign starting a sentence is simply the relative frequency of that sign appearing at the start of the sentences within the set. We thus specify the sentence model, i.e. the CPD parameters of sign-level nodes (Q_t^1 and F_t^1), by taking into account the sign sequences that appear in the training sentence set. The remaining node in the MH-HMM is the sychronization node S_t^2 whose CPD parameters are specified to implement the EX-NOR function.

After the procedure above, the MH-HMM can be used for recognition of continuously signed sentences. To recognize a test sentence, the values of all observed nodes in each time slice are input to the MH-HMM, and the most-likely sign sequence that could have produced the observed values is inferred (observed nodes in the graphical model context refers to nodes with known values). In our testing procedure, the observed nodes at time t include not just the observation features of all the components, O_t^c, for $c = 1, \ldots, N_c$, but also the nodes S_t^2 and F_t^1. As mentioned above, in order to enforce synchronization between component streams at sign boundaries, the value of the S_t^2 node must be set as 1 in all time slices. We also set $F_t^1 = 0$ for $t = 1 \ldots, T-1$ and $F_T^1 = 1$, indicating that for each test sequence, the sentence ends only at the last time slice and not before [110, 179]. This enforces the requirement that the sentence does not end until all the observations features are used up. The inferencing algorithm employed for decoding test sentences in the MH-HMM will be described in the next chapter.

4.4.2 Training H-HMMs to learn component-specific models

Our goal in this training procedure is to learn the component-specific sign and phone models, i.e. for each component c, to learn the CPD parameters for nodes

$Q_t^{2\text{-}c}$ and $F_t^{2\text{-}c}$; $Q_t^{3\text{-}c}$ and $F_t^{3\text{-}c}$; and O_t^c. Learning the CPD parameters for nodes $Q_t^{2\text{-}c}$ and $F_t^{2\text{-}c}$ requires each of their parent nodes to be in the training model. Thus, for each component c, we construct a H-HMM containing not only all the nodes above but also Q_t^1, the common parent sign node. The F_t^1 node is also included, so that we can set its value during training to indicate that for each training sequence, the sentence ends only at the last time slice. Therefore, for each sign component, c, we train a (single channel) H-HMM such as in Figure 4.8.

We denote the discrete and continuous nodes in a DBN generically as $\mathbf{Z}_t$ and $\mathbf{O}_t$. In the H-HMM of Figure 4.8, $\mathbf{O}_t$ includes just O_t^c, the vector-valued features for sign component c (in the subsequent development we drop the bold font and indicate $\mathbf{O}_t$ by O_t since there is only one continuous variable at time t). In our experiments of Chapter 6, data is obtained from direct-measure devices, and is hence insensitive to occlusion and data association ambiguity problems, and O_t^c is always observed. If the proposed model is to be used within a vision-based system, we would need to first track the required features for each component. In this case, O_t^c, would be the tracked features and not the raw observations from video data. Thus O_t^c is always observed in the sense that it is always the value of whatever the tracked feature value is at time t. All other nodes at time t are discrete and we indicate the individual nodes as $Z_{i,t}$, for $i = 1, \ldots, n_Z$. In our training procedure, besides O_t^c, there are other nodes which also have known values (i.e. are observed) during training. As explained below, F_t^1 is observed in every time slice, while Q_t^1

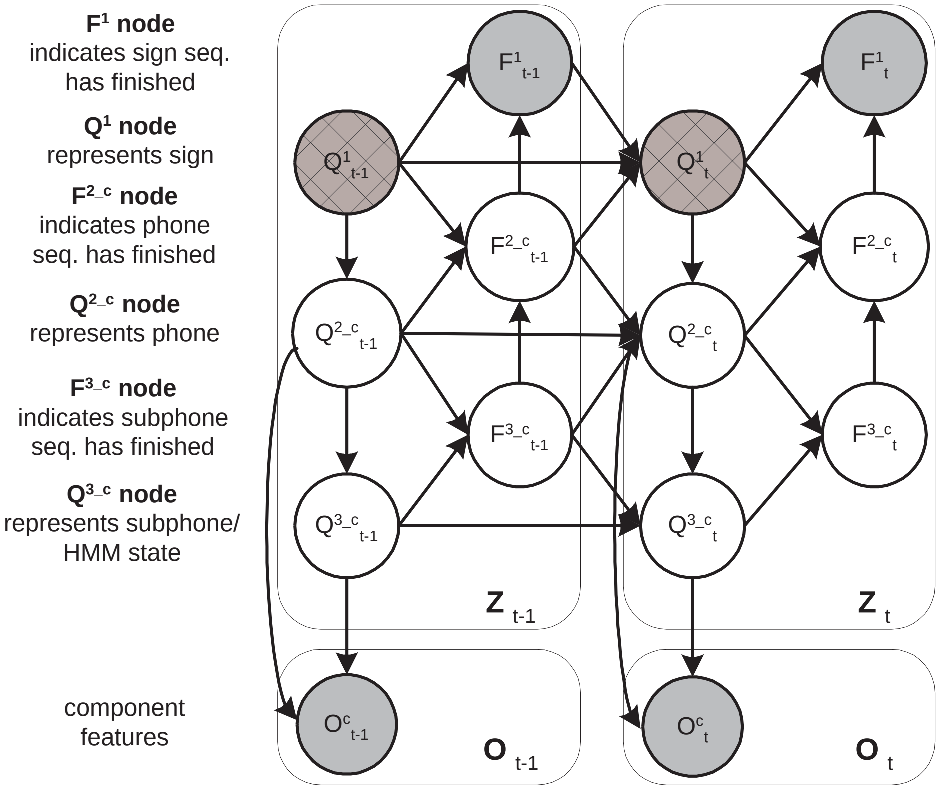


Figure 4.8: H-HMM for training sign component c. Nodes indexed by superscript c pertain to the specific component (e.g. $Q_t^{2\text{-}c}$ refers to the phone node at time t for component c). $\mathbf{Z}_t$ encompasses all discrete nodes at time t, $\mathbf{O}_t$ refers to continuous nodes, in this case just O_t^c. Solid gray nodes represent nodes that are observed in all time slices (observed nodes in the graphical model context refers to nodes whose values are known). Cross-hatched gray nodes represent nodes that are observed in some but not all time slices.

is observed only in some time slices. We denote the set of observed nodes at time t as $\mathbf{Y}_t$ and their observed values as $\mathbf{y}_t$.

Each discrete variable $Z_{i,t}$, can take on r_i possible values, $1, \ldots, r_i$. We denote the parents of each $Z_{i,t}$ as $\mathbf{Pa}_{Z_{i,t}}$. By inspection of Figure 4.8, the parents of each $Z_{i,t}$ are also discrete. We use j as the index for all possible combination of values that these parents $\mathbf{Pa}_{Z_{i,t}}$ can take and denote the index j as ranging from 1 to q_i.

Similarly, the parents $\mathbf{Pa}_{O_t}$ of continuous variable O_t are discrete, and we index the combination of their values with j, for $j = 1, \ldots, q_i$.

The parameters of the DBN are estimated with the maximum likelihood (ML) criterion, using the expectation-maximization (EM) training algorithm. This is similar to the training procedure for the BN in cases where there are missing values (unobserved nodes) in the training data. The main difference here is that in the DBN, the CPDs are time-invariant and their parameters are tied across time slices. Thus in the M-step of the EM, we not only pool together data from different training instances (in this case, training sequences) but also data from different time slices. All the terms required in the E-step can be obtained from any DBN inferencing algorithm such as the forward interface algorithm [110]. Inferencing in DBNs is covered in more detail in Chapter 5.

The training procedure for the DBN is presented in Algorithm 4.1. Each training sequence is indexed by s, for $s = 1, \ldots, N_s$, and $\mathbf{y}_{1:T}^{(s)}$[2] denotes observations from sequence s. $\boldsymbol{\theta}$ denotes all the CPD parameters of the model and $\hat{\boldsymbol{\theta}}^{(a)}$ indicates the estimated parameter values after iteration a.

Algorithm 4.1. EM algorithm for training the H-HMM

- Start with initial configurations $\hat{\boldsymbol{\theta}}^{(1)}$ for the model parameters, and iterate E

[2] In general each sequence is of different length T_s, but we drop the subscript s from T for notational simplicity.

and M-steps below until convergence.

- At iteration $a+1$,

1. **E-step**
   For $s = 1, \ldots, N_s$,
   (a) For $i = 1, \ldots, n_Z$, compute

   $$P(Z_{i,t} = k, \mathbf{Pa}_{Z_{i,t}} = j | \mathbf{y}_{1:T}^{(s)}, \hat{\boldsymbol{\theta}}^{(a)})$$

   for $t = 1, \ldots, T$; $k = 1, \ldots, r_i$; $j = 1, \ldots, q_i$. This is the joint posterior distribution of $Z_{i,t}$ and its parents $\mathbf{Pa}_{Z_{i,t}}$, given the observations $\mathbf{y}_{1:T}^{(s)}$ and the model parameters estimated at iteration a, and is the expected sufficient statistics (ESS) required for computing the parameter in Step 2(a) below [70, 110].

   (b) Compute

   $$P(\mathbf{Pa}_{O_t} = j | \mathbf{y}_{1:T}^{(s)}, \hat{\boldsymbol{\theta}}^{(a)})$$

   for $t = 1, \ldots, T$; $j = 1, \ldots, q_i$. This is the posterior distribution of $\mathbf{Pa}_{O_t}$ the parents of O_t, given the observations $\mathbf{y}_{1:T}^{(s)}$ and the model parameters estimated at iteration a, and is the ESS required for computing the parameter in Step 2(b) below [70].

   (c) Compute the likelihood for sequence s,

   $$P(\mathbf{y}_{1:T}^{(s)} | \hat{\boldsymbol{\theta}}^{(a)})$$

   required for determining convergence in Step 3 below.

2. **M-step**
   (a) The parameters for node $Z_{i,t}$'s CPD are defined as $\theta_{ijk} \triangleq P(Z_{i,t} = k | \mathbf{Pa}_{Z_{i,t}} = j)$, for $j = 1, \ldots, q_i$, and $k = 1, \ldots, r_i$. The ML estimate for θ_{ijk} at iteration $a+1$ is,

   $$\hat{\theta}_{ijk}^{(a+1)} = \frac{\sum_{s=1}^{N_s} \sum_{t=1}^{T} P(Z_{i,t} = k, \mathbf{Pa}_{Z_{i,t}} = j | \mathbf{y}_{1:T}^{(s)}, \hat{\boldsymbol{\theta}}^{(a)})}{\sum_{s=1}^{N_s} \sum_{t=1}^{T} \sum_{m=1}^{r_i} P(Z_{i,t} = m, \mathbf{Pa}_{Z_{i,t}} = j | \mathbf{y}_{1:T}^{(s)}, \hat{\boldsymbol{\theta}}^{(a)})}$$

   This is estimated for $i = 1, \ldots, n_Z$, $j = 1, \ldots, q_i$, and $k = 1, \ldots, r_i$.

(b) The CPD for O_t is a set of conditional Gaussians, $P(\underline{o}_t|\mathbf{Pa}_{O_t} = j) = \mathcal{N}(\underline{o}_t|\underline{\mu}_j, \Sigma_j)$, for $j = 1, \ldots, q_i$. The ML estimates for the Gaussian parameters at iteration $a+1$ are,

$$\hat{\underline{\mu}}_j^{(a+1)} = \frac{\sum_{s=1}^{N_s} \sum_{t=1}^{T} P(\mathbf{Pa}_{O_t} = j|\mathbf{y}_{1:T}^{(s)}, \hat{\boldsymbol{\theta}}^{(a)}) \cdot \underline{o}_t^{(s)}}{\sum_{s=1}^{N_s} \sum_{t=1}^{T} P(\mathbf{Pa}_{O_t} = j|\mathbf{y}_{1:T}^{(s)}, \hat{\boldsymbol{\theta}}^{(a)})}$$

and

$$\hat{\Sigma}_j^{(a+1)} = \frac{\sum_{s=1}^{N_s} \sum_{t=1}^{T} P(\mathbf{Pa}_{O_t} = j|\mathbf{y}_{1:T}^{(s)}, \hat{\boldsymbol{\theta}}^{(a)}) \cdot \left(\underline{o}_t^{(s)} - \hat{\underline{\mu}}_j^{(a)}\right) \left(\underline{o}_t^{(s)} - \hat{\underline{\mu}}_j^{(a)}\right)^T}{\sum_{s=1}^{N_s} \sum_{t=1}^{T} P(\mathbf{Pa}_{O_t} = j|\mathbf{y}_{1:T}^{(s)}, \hat{\boldsymbol{\theta}}^{(a)})}$$

The parameters above are estimated for $j = 1, \ldots, q_i$. $\underline{o}_t^{(s)}$ is the observed value of O_t for sequence s.

3. Stop if the incremental increase in the training data likelihood,

$$\prod_{s=1}^{N_s} P(\mathbf{y}_{1:T}^{(s)}|\hat{\boldsymbol{\theta}}^{(a)}) - \prod_{s=1}^{N_s} P(\mathbf{y}_{1:T}^{(s)}|\hat{\boldsymbol{\theta}}^{(a-1)})$$

drops below a threshold (indicating convergence), otherwise re-iterate E and M-steps.

Step 2(a) of the algorithm estimates the parameters for the CPD of discrete node $Z_{i,t}$. The summation over t is taken from 1 to T. This assumes that the CPD for node $Z_{i,t}$ is invariant for all time slices, which is the correct assumption for nodes such as F_t^1, $F_t^{2\text{-}c}$ and $F_t^{3\text{-}c}$ in the H-HMM of Figure 4.8. This is however not correct for nodes Q_t^1, $Q_t^{2\text{-}c}$ and $Q_t^{3\text{-}c}$. At time 1, all the parents (if any exist) of each of these nodes are in time 1 as well, whereas for $2 \leq t \leq T$, one or more of each node's parents are from the previous time slice. So, for example, we need to estimate the parameters of the CPD for $Q_1^{3\text{-}c}$ separately from that for $Q_t^{3\text{-}c}$ for $t = 2, \ldots, T$.

The ML estimate for parameter $P(Q_1^{3\text{-}c} = k | Q_1^{2\text{-}c} = j)$ at iteration $a+1$ is,

$$\frac{\sum_{s=1}^{N_s} P(Q_1^{3\text{-}c} = k, Q_1^{2\text{-}c} = j | \mathbf{y}_{1:T}^{(s)}, \hat{\boldsymbol{\theta}}^{(a)})}{\sum_{s=1}^{N_s} \sum_{m=1}^{r_i} P(Q_1^{3\text{-}c} = m, Q_1^{2\text{-}c} = j | \mathbf{y}_{1:T}^{(s)}, \hat{\boldsymbol{\theta}}^{(a)})}$$

The ML estimate for parameter $P(Q_t^{3\text{-}c} = k | \{Q_{t-1}^{3\text{-}c}, F_{t-1}^{3\text{-}c}, Q_t^{2\text{-}c}\} = j)$, for $t = 2, \ldots, T$, ($\{Q_{t-1}^{3\text{-}c}, F_{t-1}^{3\text{-}c}, Q_t^{2\text{-}c}\}$ indicates the combination of values for the nodes in the curly braces, and as before we index the combinations with j) at iteration $a+1$ is,

$$\frac{\sum_{s=1}^{N_s} \sum_{t=2}^{T} P(Q_t^{3\text{-}c} = k, \{Q_{t-1}^{3\text{-}c}, F_{t-1}^{3\text{-}c}, Q_t^{2\text{-}c}\} = j | \mathbf{y}_{1:T}^{(s)}, \hat{\boldsymbol{\theta}}^{(a)})}{\sum_{s=1}^{N_s} \sum_{t=2}^{T} \sum_{m=1}^{r_i} P(Q_t^{3\text{-}c} = m, \{Q_{t-1}^{3\text{-}c}, F_{t-1}^{3\text{-}c}, Q_t^{2\text{-}c}\} = j | \mathbf{y}_{1:T}^{(s)}, \hat{\boldsymbol{\theta}}^{(a)})}$$

Note that unlike at Step 2(a) above, here the summation is for $2 \leq t \leq T$ and does not include $t = 1$. With respect to the continuous node O_t^c, all its parents are in the same time slice, so that the time-invariant CPD assumption made in Step 2(b) is valid.

In the posterior distributions calculated in Step 1(a) of the algorithm, if any of $Z_{i,t}$ or $\mathbf{Pa}_{Z_{i,t}}$ is observed in the sequence s (i.e. its value is known), the observed value of the variable will have probability of one, given $\mathbf{y}_{1:T}^{(s)}$, and probabilities for all other values will be zero. For example, if $Z_{i,t}$ is observed as k_i in the sequence

s, then (conditioning on $\hat{\boldsymbol{\theta}}^{(a)}$ omitted for brevity),

$$P(Z_{i,t} = k|\mathbf{y}_{1:T}^{(s)}) = \begin{cases} 1 & \text{for } k = k_i, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the posterior probabilities in Step 1(a) are,

$$\begin{aligned} P(Z_{i,t} = k, \mathbf{Pa}_{Z_{i,t}} = j|\mathbf{y}_{1:T}^{(s)}) &= P(\mathbf{Pa}_{Z_{i,t}} = j|Z_{i,t} = k, \mathbf{y}_{1:T}^{(s)})P(Z_{i,t} = k|\mathbf{y}_{1:T}^{(s)}) \\ &= P(\mathbf{Pa}_{Z_{i,t}} = j|\mathbf{y}_{1:T}^{(s)})P(Z_{i,t} = k|\mathbf{y}_{1:T}^{(s)}) \text{ , since } Z_{i,t} \text{ is observed} \\ &= \begin{cases} P(\mathbf{Pa}_{Z_{i,t}} = j|\mathbf{y}_{1:T}^{(s)}) & \text{for } k = k_i, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{4.1}$$

A similar calculation applies to the case where one or more of the variables in $\mathbf{Pa}_{Z_{i,t}}$ is observed. In our training procedure F_t^1 is set to 0 in time slices $1\ldots, T-1$ and to 1 in time slice T, indicating that for each training sequence, the sentence ends only at the last time slice and not before [110, 179]. In Section 6.4 we explain how the value of Q_t^1 is known for some of the training sequences, and for some of the time slices in those sequences. Thus F_t^1 is observed in every time slice and every training sequence, while Q_t^1 is observed in some (but not all) time slices and in some (but not all) training sequences. Equation (4.1) can be applied when either Q_t^1 or F_t^1 appears as one of the variables in the terms calculated in Step 1(a) of the Algorithm 4.1. Since $P(Z_{i,t}, \mathbf{Pa}_{Z_{i,t}}|\mathbf{y}_{1:T}^{(s)})$ is calculated for each sequence s and each time slice t, in the case of variable Q_t^1, we simply apply Equation (4.1) if Q_t^1

happens to be observed for the sequence and time slice that we are computing the joint posterior term for.

In this training procedure we are only interested in learning the component-specific sign and phone models to be combined in the final MH-HMM. As mentioned in Section 4.4.1, the sentence model of the MH-HMM can be easily specified according to the sign sequences that appear in each training sentence. Thus the CPD parameters of sign-level nodes Q_t^1 and F_t^1 do not need to be learned during training. In our training procedure we clamp these parameters during the M-step in Algorithm 4.1, i.e. we simply skip over the variables Q_1^1, Q_t^1, and F_t^1 when estimating CPD parameters for discrete variables in Step 2(a). In addition, since for each training sequence s, we know the correct sign sequence of the sentence, we can use this knowledge of the correct sentence model for training sequence s. In other words, the CPD parameters of Q_t^1, and F_t^1 are set to values that allow the sign sequence of training sentence s to be constructed. So in the E-step of Algorithm 4.1, for each training sequence s, the CPD parameters of Q_t^1, and F_t^1 are first set to reflect the sign sequence for this training sentence before performing the computations of Step 1(a),(b) and (c). This is referred to as constrained model training and is standard practice in training procedures of speech recognition [18, 110]. Note that this is not the same as observing the node Q_t^1 – we only know the correct sign sequence of the training sentence, but not when each particular sign appears.

4.5 MH-HMM for recognition of continuous signing with inflections

This section describes how the model structure in Figure 4.7 is applied to the specific problem of recognizing continuous signing with inflections.

In our continuous signing experiments, the input ASL sentences contain signs with two types of inflections: directional verb inflections and temporal aspect inflections. As mentioned in Section 1.1.2, in directional verbs, the movement path direction serves as a pointing action which identifies the subject and the object of the verb. Section 1.1.3 described temporal aspect inflections. We will specifically consider the [DURATIONAL], [HABITUAL], and [CONTINUATIVE] aspectual inflections in our experiments. These inflections affect the movement path shape, size and speed.

Following the approach as outlined in Section 2.3, we seek to model the effect of lexical word meaning and the above inflections on the sign appearance. As mentioned in Section 1.1.2 and 1.1.3, besides movement path attributes, the inflections above also affect the location and orientation components, as follows:

- Directional verb inflections: the movement direction modulation is accompanied by a change in hand location and palm orientation.
- Temporal aspect inflections: the movement path shape and size modulations also affect the hand location.

In Chapter 3, we separated out the appearance attributes that are modulated by inflections and modelled these attributes as distinct sign components. Here however, we use the fact that the effect of the inflections above appear in both the location and orientation components to reduce the number of components that need to be modelled. Thus taking into account that lexical word meaning affects the handshape, location and orientation sign components, we find that only three components need to be modelled – handshape, location and orientation.

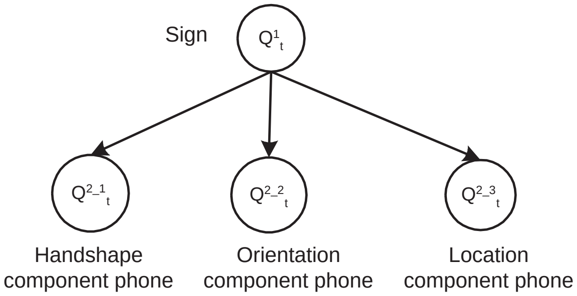


Figure 4.9: Causal dependence between the sign and the three component phone variables.

The MH-HMM structure of Figure 4.7 can thus be extended to model the three component streams, where the links between the sign-level and phone-level Q nodes at time t, are represented as in Figure 4.9. Here Q_t^1 is the sign variable, and $Q_t^{2\text{-}1}$, $Q_t^{2\text{-}2}$, $Q_t^{2\text{-}3}$ are the phone variables for the handshape component, orientation component and location component, respectively. However, the sign in fact conveys both lexical word meaning and inflectional meaning, so that we can factorize the sign variable/node Q^1 into three separate variables/nodes as:

- $Q_t^{1\text{-}LW}$: lexical word node/variable

- $Q_t^{1\text{-}DV}$: directional verb inflection node/variable
- $Q_t^{1\text{-}TA}$: temporal aspect inflection node/variable

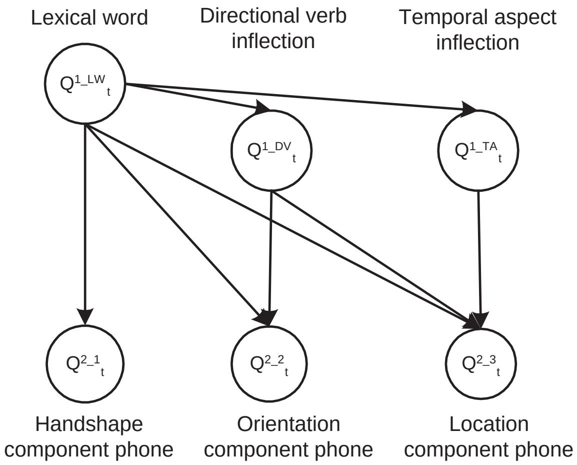

Lexical word
Directional verb inflection
Temporal aspect inflection
Q1_LWt
Q1_DVt
Q1_TAt
Q2_1t
Q2_2t
Q2_3t
Handshape component phone
Orientation component phone
Location component phone

Figure 4.10: Causal relationship between lexical word, directional verb inflections, temporal aspect inflections and the three component phone variables.

Taking into account how these top level variables affect the components of handshape, orientation and location, the links between the top level and phone-level Q nodes at time t can be represented as in Figure 4.10. Thus the Q_t^1 node of Figure 4.7 is factorized into $Q_t^{1\text{-}LW}$, $Q_t^{1\text{-}DV}$, and $Q_t^{1\text{-}TA}$ nodes as in Figure 4.10. Factorizing the sign node Q_t^1 makes clear the causal dependence between lexical root word, directional verb and temporal aspect inflections, and the three sign components of location, orientation and handshape. However, to prevent clutter we will continue using the node Q_t^1 to represent the complete sign meaning in diagrams of the MH-HMM and H-HMM structures used for training and/or continuous sign recognition. We will also often refer to the node Q_t^1 as a shorthand for the nodes $Q_t^{1\text{-}LW}$, $Q_t^{1\text{-}DV}$, and $Q_t^{1\text{-}TA}$, and the sign value as a shorthand for the combination of values in the three nodes.

The primary effect of this factorization is that it reduces the number of parameters that have to be learned for the component-specific sign models. For example, in Figure 4.10, the handshape component's phone variable only has the lexical word $Q_t^{1\text{-}LW}$ as a parent. Thus the phone sequence in the handshape component is only affected by the lexical word value and not the values of the inflection nodes. Different signs which share the same lexical meaning thus share the same phone sequence in the handshape component. This drastically reduces the number of distinct handshape-specific sign models that have to be learned in the training procedure described in Section 4.4.2. Fewer distinct models of the same type requires fewer parameters, which means that for the same amount of training data, more robust estimates for the parameters can be found. We can similarly argue for the case of the orientation component – the number of distinct orientation-specific sign models that need to be learned is reduced due to the factorization above. At first glance, the location component does not seem to benefit from this factorization. All three of the top level Q nodes are parents of the location component's phone variable. However Section 6.2 later describes how we can take advantage of context-specific independence [22] to reduce the number of distinct location-specific sign models. A second effect of factorizing the Q_t^1 node is that we can perform "partial" recognition of the sign's lexical meaning only. The complete sign meaning (i.e. the value of Q_t^1) is a combination of the values in the three nodes $Q_t^{1\text{-}LW}$, $Q_t^{1\text{-}DV}$, and $Q_t^{1\text{-}TA}$. We can recognize just the lexical/word meaning by inferring the value of $Q_t^{1\text{-}LW}$ only. Section 6.5 describes evaluation criteria that measures the sign recognition results in terms of lexical/word accuracy.

Chapter 5

Inference in dynamic Bayesian networks

We first briefly describe algorithms for exact inference in dynamic Bayesian networks (DBN) and their computational complexity. Exact inferencing is used in the E-step of the EM training algorithm described in the previous chapter. We then explain the need for applying approximate inference on the MH-HMM to recognize signs in continuous sentences. Particle filtering (PF) is proposed as a suitable approximate inference method for application to our problem domain and we show how the algorithm can be applied specifically to infer the most-probable sign sequence in a test sentence.

5.1 Exact inference in DBNs

We consider a general DBN with hidden variables $\mathbf{X}_t$, and observed variables $\mathbf{Y}_t$, at every time slice. $\mathbf{X}_t$ and $\mathbf{Y}_t$ each represent multiple variables: $\mathbf{X}_t = \{X_{1,t} \ldots X_{K,t}\}$

and $\mathbf{Y}_t = \{Y_{1,t} \dots Y_{L,t}\}$. All hidden variables are discrete. The hidden state process is first order Markov, i.e. $P(\mathbf{X}_t|\mathbf{X}_{1:t-1}) = P(\mathbf{X}_t|\mathbf{X}_{t-1})$, and the observations $\mathbf{Y}_{1:t}$, are conditionally independent given the hidden states $\mathbf{X}_{1:t}$ [37]. The graphical model representation is as in Figure 5.1.

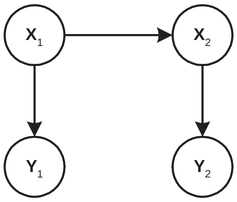


Figure 5.1: A general DBN with hidden variables $\mathbf{X}_t$, and observed variables $\mathbf{Y}_t$, unrolled for the first two time slices.

One of the goals of inferencing in DBNs is often to estimate the filtering distribution, $\alpha_{t|t}(\mathbf{x}_t) \triangleq P(\mathbf{X}_t = \mathbf{x}_t|\mathbf{y}_{1:t})$. This can be done by applying the forward-pass step of the forward-backward algorithm to DBNs [110, 131]. Given $\alpha_{t-1|t-1}(\mathbf{x}_{t-1})$, this distribution can be propagated forward to obtain $\alpha_{t|t}(\mathbf{x}_t)$ in two steps. Firstly, the one-step prediction for $\mathbf{x}_t$ is given by,

$$\begin{aligned}\alpha_{t|t-1}(\mathbf{x}_t) &\triangleq P(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \sum_{\mathbf{x}_{t-1}} P(\mathbf{x}_t|\mathbf{x}_{t-1}) P(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \\ &= \sum_{\mathbf{x}_{t-1}} P(\mathbf{x}_t|\mathbf{x}_{t-1}) \alpha_{t-1|t-1}(\mathbf{x}_{t-1}) \qquad (5.1)\end{aligned}$$

where the summation is over $\mathbf{x}_{t-1}$, the values of all the hidden variables at time $t-1$. Hence for each term $\alpha_{t|t-1}(\mathbf{x}_t)$, this step is $O(|\mathbf{X}_{t-1}|) = O(M^K)$, where $M \triangleq \max_k |X_{k,t}|$ and K is the number of hidden variables. Since there are M^K

terms $\alpha_{t|t-1}(\mathbf{x}_t)$, the total cost of this step is $O(M^{2K})$.

Next, we update the prediction using the observations at time t to obtain the posterior distribution,

$$\begin{aligned}
\alpha_{t|t}(\mathbf{x}_t) &= P(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{P(\mathbf{y}_t|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{\sum\limits_{\mathbf{x}_t} P(\mathbf{y}_t|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{y}_{1:t-1})} \\
&= \frac{P(\mathbf{y}_t|\mathbf{x}_t)\alpha_{t|t-1}(\mathbf{x}_t)}{\sum\limits_{\mathbf{x}_t} P(\mathbf{y}_t|\mathbf{x}_t)\alpha_{t|t-1}(\mathbf{x}_t)} \qquad (5.2)
\end{aligned}$$

For each term $\alpha_{t|t}(\mathbf{x}_t)$, the numerator involves one multiplication. The summation in the denominator is over $\mathbf{x}_t$ and is done just once in this step. Therefore the total cost of this step is $M^K + 1$ or $O(M^K)$. Combining the two steps, the total cost for filtering at each time slice is $O(M^{2K} + M^K)$.

When applying a DBN such as the MH-HMM (Figure 4.7) to SL recognition, the inferencing goal is to find the most-probable sign sequence in a test sentence. This amounts to finding the most-probable value assignments to a subset of the hidden variables in all the time slices (given the observations in all time slices). For example, referring to Figure 4.7, the desired inference result is $\underset{q^1_{1:T}}{\text{argmax}}\, P(Q^1_{1:T} = q^1_{1:T}|\mathbf{y}_{1:T})$. If we represent the sign variable as R_t and the other hidden variables as $\mathbf{Z}_t$, so that $\mathbf{X}_t = \{R_t, \mathbf{Z}_t\}$, the most-probable sign sequence in a data sequence with T time slices is,

$$\begin{aligned} \bar{r}_{1:T} &= \underset{r_{1:T}}{\operatorname{argmax}} P(R_{1:T} = r_{1:T} | \mathbf{y}_{1:T}) \\ &= \underset{r_{1:T}}{\operatorname{argmax}} \sum_{\mathbf{z}_{1:T}} P(R_{1:T} = r_{1:T}, \mathbf{Z}_{1:T} = \mathbf{z}_{1:T} | \mathbf{y}_{1:T}) \end{aligned} \tag{5.3}$$

This calculation can be done recursively but at a great computational cost since it involves a combination of the sum- and the max-operators which are not commutative [111]. In practice, a suboptimal solution is usually calculated, as the most-probable sequence of values for all the hidden variables rather than just the sign variable, i.e. $\underset{\mathbf{x}_{1:T}}{\operatorname{argmax}} P(\mathbf{X}_{1:T} = \mathbf{x}_{1:T} | \mathbf{y}_{1:T})$. This is suboptimal because the most-probable sequence of signs may be different from the sign sequence obtained from the most-probable sequence of all hidden variables [16].

The most-probable value assigment to all the hidden variables in all the time slices is found by replacing the sum-operator in Equation (5.1) with the max-operator [131] and keeping track of the argmax $\mathbf{x}_{t-1}$ (see Equation (5.4) below) at each time t. This max-product operation (as opposed to the sum-product operation in filtering) also has complexity $O(M^{2K} + M^K)$. The one-step prediction for $\mathbf{x}_t$ in the max-product operation is,

$$\begin{aligned}
\bar{\alpha}_{t|t-1}(\mathbf{x}_t) &\triangleq \max_{\mathbf{x}_{1:t-1}} P(\mathbf{x}_t, \mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1}) = \max_{\mathbf{x}_{t-1}} P(\mathbf{x}_t|\mathbf{x}_{t-1}) \max_{\mathbf{x}_{1:t-2}} P(\mathbf{x}_{t-1}, \mathbf{x}_{1:t-2}|\mathbf{y}_{1:t-1}) \\
&= \max_{\mathbf{x}_{t-1}} P(\mathbf{x}_t|\mathbf{x}_{t-1})\bar{\alpha}_{t-1|t-1}(\mathbf{x}_{t-1}) \qquad (5.4)
\end{aligned}$$

followed by updating the prediction using the observations at time t,

$$\begin{aligned}
\bar{\alpha}_{t|t}(\mathbf{x}_t) &\triangleq \max_{\mathbf{x}_{1:t-1}} P(\mathbf{x}_t, \mathbf{x}_{1:t-1}|\mathbf{y}_{1:t}) = \frac{P(\mathbf{y}_t|\mathbf{x}_t) \max_{\mathbf{x}_{1:t-1}} P(\mathbf{x}_t, \mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1})}{\sum_{\mathbf{x}_t} P(\mathbf{y}_t|\mathbf{x}_t) \max_{\mathbf{x}_{1:t-1}} P(\mathbf{x}_t, \mathbf{x}_{1:t-1}|\mathbf{y}_{1:t-1})} \\
&= \frac{P(\mathbf{y}_t|\mathbf{x}_t)\bar{\alpha}_{t|t-1}(\mathbf{x}_t)}{\sum_{\mathbf{x}_t} P(\mathbf{y}_t|\mathbf{x}_t)\bar{\alpha}_{t|t-1}(\mathbf{x}_t)} \qquad (5.5)
\end{aligned}$$

The E-step of the EM training algorithm in the previous chapter (refer Section 4.4.2 and Algorithm 4.1) requires calculating the joint posterior distribution of various discrete variables, given observations of all time slices (see Step 1(a) and 1(b) of Algorithm 4.1). This is a smoothing operation which requires the forward-pass mentioned above followed by a backward-pass of the same computational complexity. Thus the total cost of the smoothing operation is in general $O(2(M^{2K}+M^K))$ or $O(M^{2K}+M^K)$.

The likelihood term, $P(\mathbf{y}_{1:T})$, required in Step 1(c) of the algorithm is computed as a by-product of the forward-pass. At time t, the denominator in Equation 5.2 is $\sum_{\mathbf{x}_t} P(\mathbf{y}_t|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = P(\mathbf{y}_t|\mathbf{y}_{1:t-1})$. Collecting the denominators at $t = 1, \ldots, T$, we can calculate the likelihood as,

$$P(\mathbf{y}_{1:T}) \quad = \quad P(\mathbf{y}_1)\prod_{t=2}^{T} P(\mathbf{y}_t|\mathbf{y}_{1:t-1}) \tag{5.6}$$

Computational complexity for exact inference in DBNs has been reduced by algorithms such as the forward interface algorithm [110] which has maximum complexity of $O(M^{K+I})$, where I is the number of variables that have outgoing arcs to the next time-slice. The amount of reduction depends on the extent of inter-slice links.

5.2 Problem formulation

Our training and modelling strategy for using the MH-HMM for sign recognition requires training one H-HMM for each sign component (see Section 4.4.1). As mentioned above, the E-step of this EM training algorithm requires smoothing operations, which in the experiments of Chapter 6, was performed using the forward interface algorithm. Following training, a MH-HMM is constructed based on the CPD parameters of each H-HMM, and used to decode test sentences where the goal is to infer the most-probable sign sequence in each sentence. Time and space complexity is an issue for decoding using exact inferencing because of the large number, K, of hidden variables in the network. As mentioned above, time and space requirements are exponential in $2K$. Furthermore, examination of Figure 4.7 shows that all the hidden variables have outgoing arcs to the next time-slice, thus, although the forward interface algorithm does better than $O(M^{2K})$, the improvement is slight. Hence, it is necessary to use approximate inferencing methods to reduce time and space requirements to a manageable level.

Approximate inferencing methods that have been applied to DBNs include the Boyen-Koller algorithm [23], the factored frontier algorithm [112], loopy belief propagation [124, 113], variational algorithms [71], and stochastic (sampling) algorithms. Sampling-based algorithms have the advantage of being easy to implement on various kinds of models and giving exact answers in the limit of infinite number of samples [110]. Particle filtering (PF) is one such sampling-based method that can be applied to inferencing in DBNs.

5.3 Importance sampling and particle filtering (PF)

The most general formulation for the inference goal is the estimation of the expected value of a function of the state trajectory $\mathbf{X}_{1:t}$, or some subset of the state trajectory, relative to the posterior probability distribution (given observations), i.e. $E_{P(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})}[f(\mathbf{X}_{1:t})]$ – for example, the filtering distribution $P(\mathbf{X}_t = \mathbf{x}_t|\mathbf{y}_{1:t})$ can be expressed as $E_{P(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})}[\delta(\mathbf{X}_t, \mathbf{x}_t)]$.

The basic idea in PF is to represent the posterior $P(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})$ by samples in the state-space, for example N samples, $\mathbf{x}^i_{1:t}, i = 1 \ldots N$, and estimate the expected value of functions using these N samples instead of the exact posterior $P(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})$.

The rest of Section 5.3 is tutorial material on PF. Readers familiar with the algorithm may want to skip to the following sections.

5.3.1 Importance sampling

To estimate the expected value of some function $f(X)$ relative to $P(X)$, i.e. $E_{P(X)}[f(X)]$, the most direct method using samples draws independent identically distributed (i.i.d.) samples $x^i, i = 1 \dots N$, from $P(X)$, and estimates the required expected value as,

$$E_{P(X)}[f(X)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x^i) \quad ;\text{where } x^i \sim P(X) \tag{5.7}$$

By the law of large numbers, this estimate becomes increasingly more accurate as $N \rightarrow \infty$. In the case of the DBN models we are considering, our inference goal is to evaluate the expected value of some function of $\mathbf{X}_{1:t}$, i.e. $E_{P(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})}[f(\mathbf{X}_{1:t})]$. Usually however, it may not be feasible to sample directly from or even evaluate $P(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})$. Using the importance sampling method [36], we can instead sample from an importance function, $Q(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})$, with the requirement that $Q(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})$ dominates $P(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})$ (i.e. $Q(\mathbf{X}_{1:t}|\mathbf{y}_{1:t}) > 0$ whenever $P(\mathbf{X}_{1:t}|\mathbf{y}_{1:t}) > 0$). Often, $P(\mathbf{X}_{1:t}, \mathbf{y}_{1:t})$ is easier to evaluate than $P(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})$, thus we express $E_{P(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})}[f(\mathbf{X}_{1:t})]$ as follows,

$$\begin{aligned}
E_{P(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})}[f(\mathbf{X}_{1:t})] &= \sum_{\mathbf{x}_{1:t}} f(\mathbf{x}_{1:t})P(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) \\
&= \frac{1}{P(\mathbf{y}_{1:t})}\sum_{\mathbf{x}_{1:t}} f(\mathbf{x}_{1:t})P(\mathbf{x}_{1:t},\mathbf{y}_{1:t}) \\
&= \frac{1}{P(\mathbf{y}_{1:t})}\sum_{\mathbf{x}_{1:t}} \left\{ f(\mathbf{x}_{1:t})\frac{P(\mathbf{x}_{1:t},\mathbf{y}_{1:t})}{Q(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})}\right\} Q(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) \\
&= \frac{1}{P(\mathbf{y}_{1:t})} E_{Q(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})}[f(\mathbf{X}_{1:t})w_t(\mathbf{X}_{1:t})] \qquad (5.8)
\end{aligned}$$

where $w_t(\mathbf{X}_{1:t}) \triangleq \frac{P(\mathbf{X}_{1:t},\mathbf{y}_{1:t})}{Q(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})}$. Both the expectation term and the denominator are estimated by sampling, as shown below:

$$E_{Q(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})}[f(\mathbf{X}_{1:t})w_t(\mathbf{X}_{1:t})] \approx \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^i_{1:t})w^i_t$$

and,

$$\begin{aligned}
P(\mathbf{y}_{1:t}) &= \sum_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t},\mathbf{y}_{1:t}) \\
&= \sum_{\mathbf{x}_{1:t}} \left\{ \frac{P(\mathbf{x}_{1:t},\mathbf{y}_{1:t})}{Q(\mathbf{x}_{1:t}|\mathbf{y}_{1:t})}\right\} Q(\mathbf{x}_{1:t}|\mathbf{y}_{1:t}) \\
&= E_{Q(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})}[w_t(\mathbf{X}_{1:t})] \\
&\approx \frac{1}{N}\sum_{i=1}^{N} w^i_t
\end{aligned}$$

where $\mathbf{x}^i_{1:t} \sim Q(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})$, and $w^i_t \triangleq w_t(\mathbf{x}^i_{1:t})$ are the (unnormalised) importance weights.

Hence, combining the two terms,

$$\begin{aligned} E_{P(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})}[f(\mathbf{X}_{1:t})] &\approx \sum_{i=1}^{N} f(\mathbf{x}_{1:t}^{i}) \frac{w_t^i}{\sum_{j=1}^{N} w_t^j} \\ &= \sum_{i=1}^{N} \widetilde{w}_t^i f(\mathbf{x}_{1:t}^{i}) \end{aligned} \tag{5.9}$$

where $\widetilde{w}_t^i = \frac{w_t^i}{\sum_{j=1}^{N} w_t^j}$ are the normalised importance weights.

So basically, by drawing samples $\mathbf{x}_{1:t}^{i}$ from $Q(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})$, the expected value of any function $f(\mathbf{X}_{1:t})$ can be evaluated relative to the distribution $P(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})$, as a weighted sum of the function evaluated at $\mathbf{x}_{1:t}^{i}$, with weights $\widetilde{w}_t^i$ defined as above.

For example, we can estimate the posterior distribution $P(\mathbf{X}_{1:t} = \mathbf{x}_{1:t}|\mathbf{y}_{1:t})$ itself as,

$$\begin{aligned} P(\mathbf{X}_{1:t} = \mathbf{x}_{1:t}|\mathbf{y}_{1:t}) &= E_{P(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})}[\delta(\mathbf{X}_{1:t}, \mathbf{x}_{1:t})] \\ &\approx \sum_{i=1}^{N} \widetilde{w}_t^i \delta(\mathbf{x}_{1:t}^{i}, \mathbf{x}_{1:t}) \end{aligned} \tag{5.10}$$

where $\delta(x, x') = 1$ if $x = x'$, and 0 otherwise.

5.3.2 Sequential importance sampling

In practice, it is not necessary to sample the entire trajectory $\mathbf{x}_{1:t}^{i}$, N times from $Q(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})$. Since at time $t-1$, we have N trajectory samples $\mathbf{x}_{1:t-1}^{i}$, sampled from

$Q(\mathbf{X}_{1:t-1}|\mathbf{y}_{1:t-1})$, to represent $P(\mathbf{X}_{1:t-1}|\mathbf{y}_{1:t-1})$, we can propagate the N trajectories to time t to obtain N samples $\mathbf{x}^i_{1:t}$, sampled from $Q(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})$, without modifying the previous simulated trajectories. This means that the importance function at time t, $Q(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})$, admits as a marginal distribution, the importance function $Q(\mathbf{X}_{1:t-1}|\mathbf{y}_{1:t-1})$ at time $t-1$ [36],

$$Q(\mathbf{X}_{1:t}|\mathbf{y}_{1:t}) = Q(\mathbf{X}_t|\mathbf{X}_{1:t-1}, \mathbf{y}_{1:t})Q(\mathbf{X}_{1:t-1}|\mathbf{y}_{1:t-1}) \tag{5.11}$$

We thus incrementally sample from $Q(\mathbf{X}_t|\mathbf{X}_{1:t-1}, \mathbf{y}_{1:t})$ at every time step. In effect, we are choosing importance functions that are conditionally independent of observations in the future, $Q(\mathbf{X}_{1:t}|\mathbf{y}_{1:t+k}) = Q(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})$. With this choice, the weights w^i_t can also be evaluated incrementally. Using the definition of w^i_t from equation (5.8), we have,

$$\begin{aligned}
w^i_t &= \frac{P(\mathbf{x}^i_{1:t}, \mathbf{y}_{1:t})}{Q(\mathbf{x}^i_{1:t}|\mathbf{y}_{1:t})} \\
&= \frac{P(\mathbf{x}^i_t, \mathbf{y}_t|\mathbf{x}^i_{1:t-1}, \mathbf{y}_{1:t-1})}{Q(\mathbf{x}^i_t|\mathbf{x}^i_{1:t-1}, \mathbf{y}_{1:t})} \cdot \frac{P(\mathbf{x}^i_{1:t-1}, \mathbf{y}_{1:t-1})}{Q(\mathbf{x}^i_{1:t-1}|\mathbf{y}_{1:t-1})} \\
&= \frac{P(\mathbf{y}_t|\mathbf{x}^i_{1:t}, \mathbf{y}_{1:t-1})P(\mathbf{x}^i_t|\mathbf{x}^i_{1:t-1}, \mathbf{y}_{1:t-1})}{Q(\mathbf{x}^i_t|\mathbf{x}^i_{1:t-1}, \mathbf{y}_{1:t})} \cdot w^i_{t-1} \\
&= \frac{P(\mathbf{y}_t|\mathbf{x}^i_t)P(\mathbf{x}^i_t|\mathbf{x}^i_{t-1})}{Q(\mathbf{x}^i_t|\mathbf{x}^i_{1:t-1}, \mathbf{y}_{1:t})} \cdot w^i_{t-1}
\end{aligned} \tag{5.12}$$

5.3.3 Sequential Importance Sampling with Resampling

Doucet [36] shows that for importance functions satisfying equation (5.11) above, the variance of the importance weights increases stochastically with time. This implies that after a few steps of sequential sampling, most of the normalized importance weights will be very close to zero. Thus much of the computation is spent on updating sample trajectories which will finally contribute very little to the posterior distribution estimate. Resampling is a method to counter the degeneracy of importance weights by eliminating trajectories with small values of normalized importance weights and replicating trajectories with large values. In Sampling Importance Resampling (SIR) [132], each trajectory sample is replicated with probability proportional to its normalized weight. This amounts to sampling with replacement from the current belief state, since the weight and frequency of particles reflect that belief state. After resampling, all the samples are equally weighted. In the large sample limit, the representation of the posterior distribution remains unchanged after resampling [133].

The particle filtering algorithms most often in use resample at every time slice, as described in Algorithm 5.1.

Algorithm 5.1. Particle Filtering or Sequential Importance Sampling with Resampling

```
1. Sequential Importance Sampling step
    (a) For $i = 1 \ldots N$, obtain samples (equation (5.11))
```

$$\hat{\mathbf{x}}_t^i \sim Q(\mathbf{X}_t|\mathbf{x}_{1:t-1}^i, \mathbf{y}_{1:t})$$

and set

$$\hat{\mathbf{x}}_{1:t}^i \triangleq (\hat{\mathbf{x}}_t^i, \mathbf{x}_{1:t-1}^i)$$

(b) For $i = 1 \ldots N$, evaluate importance weights up to a normalizing constant (equation (5.12))

$$w_t^i \propto \frac{P(\mathbf{y}_t|\mathbf{x}_t^i)P(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{Q(\mathbf{x}_t^i|\mathbf{x}_{1:t-1}^i, \mathbf{y}_{1:t})}$$

(c) For $i = 1 \ldots N$, normalize the importance weights

$$\widetilde{w}_t^i = \frac{w_t^i}{\sum_{j=1}^N w_t^j}$$

2. Resampling step

   - Resample N samples from $\hat{\mathbf{x}}_{1:t}^i$ according to the normalized importance weights $\widetilde{w}_t^i$, to obtain N samples $\mathbf{x}_{1:t}^i$.

Note that in Step 1(b) of Algorithm 5.1, the expression shown for w_t^i is actually the incremental weight at time t (equation (5.12)). However since all weights were set to be equal after resampling at the previous time slice, this has no bearing on the final normalized weight values. We next analyze two choices of importance functions that satisfy equation (5.11), and the associated importance weights.

5.3.4 Importance function and importance weights

Prior importance function

The simplest importance function for sampling $\mathbf{X}_t$ is the prior distribution of the hidden state variables, $P(\mathbf{X}_t|\mathbf{x}_{1:t-1}^i, \mathbf{y}_{1:t-1})$ [36]. This is the distribution of the hidden state $\mathbf{X}_t$, given a past sample trajectory and past observations, and before seeing the current observations $\mathbf{y}_t$. In general, when there are multiple hidden variables, $\mathbf{X}_t = \{X_{1,t} \ldots X_{K,t}\}$, we sample each $X_{k,t}$ in topological order such that the parents of $X_{k,t}$ in the current time slice are always sampled before it. Since the parents of $X_{k,t}$ from the previous time slice have already been sampled, the sampling distribution for $X_{k,t}$ is just the distribution defined by its local conditional probability (CPD) and by the values of its parents. Thus the prior importance function is expressed as,

$$\begin{aligned} Q(\mathbf{X}_t|\mathbf{x}^i_{1:t-1}, \mathbf{y}_{1:t}) &= P(\mathbf{X}_t|\mathbf{x}^i_{1:t-1}, \mathbf{y}_{1:t-1}) = P(\mathbf{X}_t|\mathbf{x}^i_{t-1}) \quad \text{(due to Markov state process)} \\ &= \prod_{k=1}^{K} P(X_{k,t}|\mathbf{Pa}_{X_{k,t}} = \mathbf{pa}^i_{X_{k,t}}) \qquad (5.13) \end{aligned}$$

where $P(X_{k,t}|\mathbf{Pa}_{X_{k,t}})$ is the CPD of $X_{k,t}$, and $\mathbf{Pa}_{X_{k,t}}$ are the parents of $X_{k,t}$ with instantiated values $\mathbf{pa}^i_{X_{k,t}}$. The $X_{k,t}$ variables are sampled one at a time, and it is never necessary to evaluate the full joint prior, $P(\mathbf{X}_t|\mathbf{x}^i_{t-1})$. For each sample i, we need to sample once per state variable, $X_{k,t}$. The appropriate sampling distribution is found by indexing into $X_{k,t}$'s CPD using the instantiated values of its parents. This indexing requires number of operations in the order of the number of parents, $N_{\mathbf{Pa}_{X_{k,t}}}$. Since all the $X_{k,t}$ variables are discrete, the CPD indexed by the instantiated values of the parents is a multinomial distribution. Sampling from a multinomial distribution with m values requires $O(\log m)$ operations [84]. Thus, denoting $N_P \triangleq \max_k[N_{\mathbf{Pa}_{X_{k,t}}}]$, and $M \triangleq \max_k|X_{k,t}|$, the overall cost of each sample is $O(K \times N_P \times log(M))$, which is linear in K, the number of hidden variables.

With this choice of importance function, the importance weights correspond to the observation likelihood. From equation (5.12) we have,

$$\begin{aligned} w_t^i &\propto \frac{P(\mathbf{y}_t|\mathbf{x}_t^i)P(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{Q(\mathbf{x}_t^i|\mathbf{x}_{1:t-1}^i, \mathbf{y}_{1:t})} \\ &= \frac{P(\mathbf{y}_t|\mathbf{x}_t^i)P(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{P(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)} = P(\mathbf{y}_t|\mathbf{x}_t^i) \triangleq l_{t|t}^i \end{aligned} \tag{5.14}$$

When there are multiple variables in $\mathbf{Y}_t$, say $Y_{l,t}, l = 1 \ldots L$, the likelihood is evaluated as,

$$l_{t|t}^i = \prod_{l=1}^{L} P(y_{l,t}|\mathbf{Pa}_{Y_{l,t}} = \mathbf{pa}_{Y_{l,t}}^i) \tag{5.15}$$

where $P(y_{l,t}|\mathbf{Pa}_{Y_{l,t}})$ is the CPD of $Y_{l,t}$ evaluated at $y_{l,t}$, and with instantiated parent values $\mathbf{pa}_{Y_{l,t}}^i$. All parents of $Y_{l,t}$ are instantiated because all the hidden variables $X_{k,t}$ have already been sampled at this point. The evaluation of the importance weights is thus linear in L, the number of observation variables.

Optimal importance function

The optimal importance function in terms of minimizing the variance of the (un-normalized) importance weights w_t^i, conditioned on the sample trajectory $\mathbf{x}_{1:t-1}^i$ and observations $\mathbf{y}_{1:t}$, is the posterior distribution of the hidden state variables,

$P(\mathbf{X}_t|\mathbf{x}^i_{1:t-1},\mathbf{y}_{1:t})$ [36] (recall that low variance is a desirable property for reducing degeneracy in the normalized importance weights). Thus the optimal importance function is evaluated as,

$$\begin{aligned}
Q(\mathbf{X}_t|\mathbf{x}^i_{1:t-1},\mathbf{y}_{1:t}) &= P(\mathbf{X}_t|\mathbf{x}^i_{1:t-1},\mathbf{y}_{1:t}) = P(\mathbf{X}_t|\mathbf{x}^i_{t-1},\mathbf{y}_t) \quad \text{(due to Markov state process)} \\
&= \frac{P(\mathbf{y}_t|\mathbf{X}_t)P(\mathbf{X}_t|\mathbf{x}^i_{t-1})}{\sum_{\mathbf{x}_t} P(\mathbf{y}_t|\mathbf{X}_t)P(\mathbf{X}_t|\mathbf{x}^i_{t-1})} \\
&= \frac{\prod_{l=1}^{L} P(y_{l,t}|\mathbf{Pa}_{Y_{l,t}}) \prod_{k=1}^{K} P(X_{k,t}|\mathbf{Pa}_{X_{k,t}})}{P(\mathbf{y}_t|\mathbf{x}^i_{t-1})} \qquad (5.16)
\end{aligned}$$

where $P(X_{k,t}|\mathbf{Pa}_{X_{k,t}})$ is the CPD of $X_{k,t}$, and $P(y_{l,t}|\mathbf{Pa}_{Y_{l,t}})$ is the CPD of $Y_{l,t}$ evaluated at $y_{l,t}$. The sampling distribution has to be evaluated $|\mathbf{X}_t|$ times, so if there are K variables in $\mathbf{X}_t$ and as before $M \triangleq \max_k |X_{k,t}|$, the cost of each sample is $O(M^K)$, i.e. exponential in K. With this choice of importance function, the importance weights correspond to the one-step ahead observation likelihood. From equation (5.12) we have,

$$\begin{aligned}
w^i_t &\propto \frac{P(\mathbf{y}_t|\mathbf{x}^i_t)P(\mathbf{x}^i_t|\mathbf{x}^i_{t-1})}{Q(\mathbf{x}^i_t|\mathbf{x}^i_{1:t-1},\mathbf{y}_{1:t})} \\
&= P(\mathbf{y}_t|\mathbf{x}^i_t)P(\mathbf{x}^i_t|\mathbf{x}^i_{t-1}) \cdot \frac{P(\mathbf{y}_t|\mathbf{x}^i_{t-1})}{P(\mathbf{y}_t|\mathbf{x}^i_t)P(\mathbf{x}^i_t|\mathbf{x}^i_{t-1})} \\
&= P(\mathbf{y}_t|\mathbf{x}^i_{t-1}) \triangleq l^i_{t|t-1} \qquad (5.17)
\end{aligned}$$

The one-step ahead observation likelihood $l^i_{t|t-1}$ is calculated at the same time as the importance function since it appears in the denominator in equation (5.16).

Thus the calculation complexity is also exponential in K.

5.4 Comparison of computational complexity

Table 5.1 compares the computational complexity of exact filtering and PF (with different choices of importance functions) in DBN.

	Cost per time step	
Exact filtering	$O(M^{2K} + M^K)$	
	Cost per time step	
	Sampling	Weights
PF with prior importance function	$O(SK)$	$O(SL)$
PF with optimal importance function	$O(SM^K)$	$O(SM^K)$
Notations: $M \triangleq \max_k \lvert X_{k,t} \rvert$; K = total number of hidden variables, $\mathbf{X}_t$ L = total number of observation variables, $\mathbf{Y}_t$ S = number of samples		

Table 5.1: Computational complexity for exact and approximate (sampling) inferencing in DBN

In terms of dependence on K, the number of hidden variables, PF with prior sampling provides substantial computational saving as compared to exact inferencing and even PF with optimal sampling. The latter two methods are exponential in K, while PF with prior sampling is linear in K. However, PF methods are also linear in S, the number of samples employed. The samples represent the posterior distribution $P(\mathbf{X}_{1:t}|\mathbf{y}_{1:t})$, therefore the larger the state space, M^K, the more samples are required to represent the distribution at an acceptable accuracy. In

Chapter 6 we will use PF with prior sampling to perform inferencing on MH-HMM models and investigate the effects of sample number on sign recognition accuracy.

5.5 Continuous sign recognition using PF

In continuous sign recognition, the goal of inferencing is to find the most-probable sign sequence. The approach required is to marginalize away the non-sign hidden variables in the model (for example the phone, subphone and indicator variables in the MH-HMM), before maximizing the sequence of sign values. As mentioned in Section 5.1, in practice, the suboptimal solution of the most-probable sequence of values for all the hidden variables is usually calculated instead.

With sampling methods like PF however, it is relatively straightforward to estimate the most-probable sequence of sign values. It simply involves counting sample trajectories.

Representing the sign variable as R_t and the other hidden variables as $\mathbf{Z}_t$, so that $\mathbf{X}_t = \{R_t, \mathbf{Z}_t\}$, the most-probable sign sequence in a data sequence with T time slices is $\underset{r_{1:T}}{\operatorname{argmax}} P(R_{1:T} = r_{1:T}|\mathbf{y}_{1:T})$. We can estimate the posterior distribution $P(\mathbf{X}_{1:T} = \mathbf{x}_{1:T}|\mathbf{y}_{1:T})$ from sample trajectories $\mathbf{x}^i_{1:T}, i = 1 \ldots N$ using equation (5.10). Basically, a weighted sum of the samples $\mathbf{x}^i_{1:T}$ is calculated for each of the values $\mathbf{x}_{1:T}$ whose posterior probability is required. The most-probable value of $\mathbf{x}_{1:T}$ is the one with the largest weighted sum. Similarly, we can estimate the marginal posterior distribution $P(R_{1:T} = r_{1:T}|\mathbf{y}_{1:T})$ by calculating a weighted sum of the samples $\mathbf{x}^i_{1:T}$ where the $R_{1:T}$ variables in the sample take on each of the values $r_{1:T}$ whose posterior probability we want to evaluate.

$$\begin{aligned}
P(R_{1:T} = r_{1:T}|\mathbf{y}_{1:T}) &= \sum_{\mathbf{z}_{1:T}} P(R_{1:T} = r_{1:T}, \mathbf{Z}_{1:T} = \mathbf{z}_{1:T}|\mathbf{y}_{1:T}) \\
&= \sum_{\mathbf{z}_{1:T}} P(\mathbf{X}_{1:T} = \mathbf{x}_{1:T}|\mathbf{y}_{1:T}) \text{ , since } \mathbf{X}_t = \{R_t, \mathbf{Z}_t\} \\
&\approx \sum_{\mathbf{z}_{1:T}} \sum_{i=1}^{N} \widetilde{w}^i_T \delta(\mathbf{x}^i_{1:T}, \mathbf{x}_{1:T}) \text{ , from equation (5.10)} \\
&= \sum_{i=1}^{N} \widetilde{w}^i_T \sum_{\mathbf{z}_{1:T}} \delta(\mathbf{x}^i_{1:T}, \mathbf{x}_{1:T}) \\
&= \sum_{i=1}^{N} \widetilde{w}^i_T \sum_{\mathbf{z}_{1:T}} \delta(\{r^i_{1:T}, \mathbf{z}^i_{1:T}\}, \{r_{1:T}, \mathbf{z}_{1:T}\}) \\
&= \sum_{i=1}^{N} \widetilde{w}^i_T \delta(r^i_{1:T}, r_{1:T}). \qquad (5.18)
\end{aligned}$$

So to evaluate the posterior probability of a particular sign sequence, $P(R_{1:T} = r_{1:T}|\mathbf{y}_{1:T})$, we do a weighted sum of the sample trajectories, $\mathbf{x}^i_{1:T}$, where the sampled values for the variables $R_{1:T}$ is the same as the sign sequence whose probability we need to estimate. The most-probable value of $r_{1:T}$ is the one with the largest weighted sum. This is a slightly different application of the PF algorithm from what is usually found in the literature, which generally estimates the filtering distribution. Our application is suitable for the case where we are interested in the values of only a subset of the hidden variables and we want the most-probable sequence of values for this subset of hidden variables.

Chapter 6

Experimental results

In this chapter we present experimental results on recognizing continuously sentences that include inflected signs, using the model proposed in Chapter 4. Inferencing on this model employs the PF algorithm outlined in Chapter 5. We first describe the data collection process including the sign vocabulary, and feature extraction for each sign component. Section 6.2 describes how we obtain initial parameters for training component-specific sign and phone models. In this section we first review how sign and phone models have been defined in previous work before presenting our approach to this issue. In Section 6.3 we review some of the past work on dealing with movement epenthesis before presenting our approach. Section 6.4 discusses the possible advantages to be gained by labelling sign node values in the training data. This is followed by sections on the evaluation criteria for test results and the presentation of those results. The most important results are in Sections 6.7 and 6.8. Sections 6.7 presents results on recognizing continuous signs by combining information from multiple sign components. Section 6.8 presents a procedure whereby we perform training on sentences containing only a subset of signs in the vocabulary, and subsequently use the trained model to recognize sentences containing unseen signs. PF is used as the inferencing algorithm in both cases and we present experiment results using different numbers of samples in the algorithm.

6.1 Data collection

6.1.1 Sign vocabulary and sentences

The collected data is obtained from a deaf individual who is a native signer of the local (Singaporean) sign language. The signed sentences, which adhered to ASL grammar, were continuous, with no pauses between signs. There were 73 distinct sentences between 2 to 6 signs long, constructed from a 98-sign vocabulary. Each distinct sentence was signed approximately 5 times, providing a total of 343 sentences and 1927 signs. The 98-sign vocabulary includes signs with inflections, specifically, directional verb inflections and temporal aspect inflections (as described in Sections 1.1.2 and 1.1.3). Such inflected signs are formed from a combination of a root lexical word and one or more inflection values. The vocabulary included both one-handed and two-handed signs. However all the signs were distinguishable by looking only at the dominant hand, i.e. no two signs in the vocabulary had exactly the same appearance in the dominant hand, and differed only in the appearance of the non-dominant hand. Table B.1 in Appendix B lists the 29 different lexical words present in the vocabulary. There are three different temporal aspect inflection values (see Table B.2 in Appendix B) and 11 different directional verb inflection values (see Table B.3 in Appendix B) that may combine with a root lexical word.

Examples of directional verb and temporal aspect inflected signs in the vocabulary are given below:

- The root verb HELP, combined with inflection values indicating different subjects and objects, yields: $\text{HELP}^{\text{I}\rightarrow\text{YOU}}$, $\text{HELP}^{\text{YOU}\rightarrow\text{I}}$, $\text{HELP}^{\text{I}\rightarrow\text{GIRL}}$, $\text{HELP}^{\text{I}\rightarrow\text{JOHN}}$, $\text{HELP}^{\text{JOHN}\rightarrow\text{I}}$, $\text{HELP}^{\text{JOHN}\rightarrow\text{YOU}}$, $\text{HELP}^{\text{YOU}\rightarrow\text{HELP}}$, $\text{HELP}^{\text{GIRL}\rightarrow\text{I}}$, $\text{HELP}^{\text{GIRL}\rightarrow\text{YOU}}$, $\text{HELP}^{\text{YOU}\rightarrow\text{GIRL}}$, $\text{HELP}^{\text{GIRL}\rightarrow\text{JOHN}}$.

- The root word EAT, combined with different temporal aspect inflections yields:
  $\text{EAT}^{[\text{DURATIONAL}]}$, $\text{EAT}^{[\text{HABITUAL}]}$, $\text{EAT}^{[\text{CONTINUATIVE}]}$.

Some of the inflected signs are formed with two inflection values which appear simultaneously, further increasing the complexity of the vocabulary. Examples of these signs are: $(\text{GIVE}^{[\text{DURATIONAL}]})^{\text{I}\rightarrow\text{GIRL}}$, $(\text{GIVE}^{[\text{HABITUAL}]})^{\text{I}\rightarrow\text{GIRL}}$, $(\text{GIVE}^{[\text{CONTINUATIVE}]})^{\text{I}\rightarrow\text{GIRL}}$

A few of the lexical root words are used in combination with various inflection values to form many different signs, for example, the lexical word GIVE appears in 16 different signs.

6.1.2 Data measurement and feature extraction

Features were extracted from the signer's right (dominant) hand only[1]. Data was collected using the Polhemus electromagnetic tracker [2] which consists of an electromagnetic field-emitting transmitter and sensors that detect their 3-dimensional position and orientation within the field. Sensors were placed on the back of the signer's right hand and the base of his spine. Conceptually, each sensor has an attached orthogonal coordinate frame. The position and orientation of the right hand's sensor is represented by the 3-dimensional coordinates of its origin, x, y, and z axes ($\underline{o}_H$, $\underline{x}_H$, $\underline{y}_H$, and $\underline{z}_H$), relative to the waist sensor's coordinate frame. Appendix C gives details of how this is calculated. The waist sensor's coordinate frame was used as a reference to discount variations in the signer's position and the direction he is facing, relative to the transmitter. We also collected data from a Virtual Technologies Cyberglove [4] worn on the right hand. This records the fingers' joint and abduction angles, and the wrist pitch and yaw, from 18 sensors in the glove. The tracker and glove data are synchronized and were recorded at approximately 31.1ms frame rate.

[1] The sign vocabulary included two-handed signs. Since only features from the right hand are extracted in our experiments, any potential information conveyed by the left (non-dominant) hand in two-handed signs is ignored. The implications of this are discussed in the next chapter, Section 7.2.

As mentioned in Section 4.5, in the continuous signing experiments, we model three sign components – handshape, location and orientation. The features used as observations for each of the components are given below:

- Handshape component. Data measured by 16 sensors of the Cyberglove, reporting the joint and abduction angles of the right hand's fingers and thumb. The data reported by the two sensors measuring wrist yaw and pitch were not used because this data does not represent the finger configurations. The feature vector for the handshape component is 16-dimensional.

- Location component. The 3-dimensional position of the right hand, $\underline{o}_H$, taken to be the origin of the sensor's coordinate frame. The feature vector for the location component is 3-dimensional.

- Orientation component. The unit vector corresponding to the z-axis, $\underline{z}_H$, of the right hand sensor, with reference to the waist sensor's coordinate frame. Recall from Section 1.1.1 that the hand orientation is defined as the direction in which the palm and fingers are pointing. Here however, we only extract features measuring the palm direction because measurements pertaining to the fingers are already extracted in the feature vector of the handshape component. Figure 6.1 shows a schematic of how the sensor is mounted on the back of the right hand. The x, y and z-axes of the right hand sensor's coordinate frame are shown. The sensor's z-axis direction is roughly coincident with the direction in which the palm is pointing thus its corresponding unit vector indicate the palm orientation. We note that left-right rotation (i.e. hand rotations in the x-y plane) would not register a change in the z-axis direction. So our choice of features is based on a simplifying assumption that the direction in which the palm is pointing is more relevant than the left-right wrist rotation. The feature vector for the orientation component is 3-dimensional.

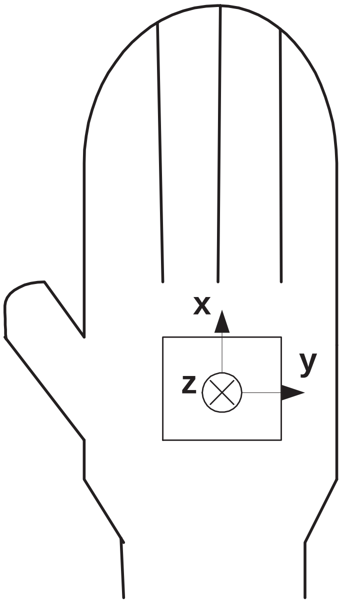


Figure 6.1: Schematic representation of how the Polhemus tracker sensor is mounted on the back of the right hand. The z-axis of the sensor's coordinate frame is pointing into the page, i.e. it is approximately coincident with the direction that the palm is facing.

6.2 Initial parameters for training component-specific models

One of the difficulties faced by researchers in SL recognition who wish to take the approach of modelling subunits or phones is the lack of a general consensus in SL linguistic studies as to what those subunits are. Sign linguists do agree that a sign consists of parts and that each of these parts has a limited number of categories or“primes”. A SL recognition researcher may want to equate phones with these primes since the goal is to decompose a sign into a limited number of phones. However, there is no consensus among SL linguists as to how many primes exist, for example, various numbers of distinct handshapes have been proposed, such as 19, 40, 45 and 54 [10]. Although there has been previous SL recognition work [157, 161, 174] that define sign subunits linguistically, in these works, the analysis and definition of subunits/phones is based on a particular phonological model proposed by SL linguists and not a commonly agreed upon model. Furthermore, there may be a mismatch between the phonological model employed and the observation feature vectors that have been found to be the most robust for recognition. For example, Vogler [157] defined subunits based on Liddell's Movement-Hold (M-H) phonological model [95]. In the M-H framework, translational movement of the hand tracing the same trajectory shape in space and moving in roughly the same direction are defined as the same phone, regardless of the height at which the movement is performed. For example, "$M - \{str_{Toward}\}$" (a straight line movement towards the body) performed at the chest, chin or forehead-level are defined as equivalent phones. Thus the phone appears to be position-invariant. This is not a realistic match with the HMM phone models as defined by Vogler in [157] where the observation features include the 3-dimensional positional data of the hand, which is not position-invariant. Although the 3-dimensional hand velocity would be a good candidate as a position-invariant feature, it was found to be susceptible to noise and yielded comparatively poor recognition results.

Alternative data-driven approaches are based on clustering the data. Based on unsupervised methods employed in speech recognition [69], Bauer and Kraiss [12] defined 10 subunits for a vocabulary of 12 signs using k-means clustering. The data was obtained from all time slices of a sentence and is clustered in a feature space that is a concatenation of measurements from the sign components of hand location, orientation and handshape. Continuous sentences need to be manually segmented in time into the constituent signs so that a particular sign can be defined as a sequence of the subunits found through clustering. Fang et. al [43] used temporal clustering to extract subunits by first segmenting a sentence using HMMs, then clustering the segments by using dynamic programming to compute distance measures. Each segment consists of a sequence of concatenated features of hand location, orientation and handshape. Wang et. al [160] found handshape phones by clustering handshape features only, using a combination of Kohonen's

SOM and k-means. However they did not show any sign-level recognition results using the proposed phones.

Similar to Bauer and Kraiss [12], our approach for obtaining phone models is also based on clustering but differs from their work and Fang 's [43] in that the phones are defined separately for each of the sign components; and unlike [160] we obtain phones for the location and orientation components, as well as handshape.

The 343 sentences (containing 1927 signs) collected were first divided into training and test sets in the ratio of approximately 60:40, resulting in 201 training sentences (containing 1139 signs), and 142 test sentences (containing 788 signs). Then for each of the distinct signs that appear in the vocabulary, we found one sentence containing the sign from the training set and manually segmented the sentence in time into its constituent signs. Manual segmentation of the glove and tracker data was performed by determining sign boundaries through inspection of video sequences of the signer that were recorded simultaneously, and then calculating the closest data frames corresponding to those boundaries. The correspondence between video and data frames is not exact because of their different frame rates (video frame rate 33.3ms, data frame rate 31.1ms). In total 67 sentences out of the 201 training sentences, were processed as above, i.e. approximately $\frac{1}{3}$ of the training set.

We then performed clustering in the feature space corresponding to each of the sign components of location, orientation and handshape (see Section 6.1.2). Since we are interested in finding subunits which define signs, we only clustered data from time slices corresponding to when signs occur in the sentence, discarding time slices corresponding to movement epenthesis (see Section 1.1.1 for a description of movement epenthesis). So for example, to obtain location phone models, we take the $\underline{o}_H$ feature vector of each data frame that corresponds to valid signs from the manually segmented sentences, and perform k-means clustering. That is, the frame-by-frame feature vectors corresponding to valid signs are clustered. An initial guess of the number of clusters is based on a ballpark range of the number of phones proposed in the sign linguistic literature. Subsequently, clusters with fewer than two members were merged with the nearest neighbouring cluster. We arrived at 28 clusters for location and 40 each for handshape and orientation.

There are two main purposes for the clustering procedure above. Firstly, it defines the number of phones in each component (thus there are 28 location phones, and 40 each of handshape and orientation phones) and provides initial parameters for the phone models in the EM training algorithm (Algorithm 4.1). Recall from Section 4.4.2 that the EM training algorithm is used to learn the component-specific sign and phone models. The EM algorithm requires initialization of all the parameters in the model to be trained. These parameters should be well-chosen as the algorithm finds only local maxima and is thus sensitive to the choice of initial starting point. The CPD parameters pertaining to the sentence model (i.e.

CPD parameters for nodes Q_t^1 and F_t^1 in the H-HMM of Figure 4.8) do not require initialization since their values are adjusted for each training sentence to reflect the correct sign sequence for that sentence. The CPD parameters pertaining to component-specific phone models do need to be initialized. The initial parameters for CPDs of the $Q_t^{3\text{-}c}$ and $F_t^{3\text{-}c}$ nodes are generally set to define subphone state transitions following the 3-state left-right (Bakis) model (see Figure 4.2), with equal probabilities specified for state transitions with non-zero values. In Section 6.6 the means and covariances of the clusters found above are used to initialize the CPD parameters of O_t^c.

This brings us to the second purpose of the clustering procedure which is to obtain initial parameters for the component-specific sign models in the EM training algorithm, as explained in the following paragraphs.

Our approach to modelling a sign is to define it as consisting of synchronized sequences of distinct values or phones in each sign component (see Section 2.2). Thus in our next step, we obtain for each of the distinct signs in the vocabulary, initial sign models (i.e. phone sequences) in each of the sign components. The component-specific phone sequence for a particular sign is the sequence of cluster assignments for the appropriate data features (according to the sign component) corresponding to the time slices for that sign. In other words, it is the winning cluster sequence for the component features in that sign. The phone sequences obtained are then used to initialize the CPD parameters that define the component-specific sign models in the H-HMM of Figure 4.8 – specifically, the CPD parameters of the $Q_t^{2\text{-}c}$ and $F_t^{2\text{-}c}$ nodes which encode the sign models for the c component. A particular phone sequence is completely specified by the state initial, transition and ending probabilities at the phone-level. The state initial and transition probabilities at the phone-level are encoded in the CPD parameters of $Q_t^{2\text{-}c}$, while the ending probabilities are encoded in the CPD parameters of $F_t^{2\text{-}c}$. Thus the phone sequences found above can be used to provide initial values for these parameters.

In our implementation, there are fewer than 98 distinct sign models in each component as one would expect for a 98-sign vocabulary. This is due to the causal dependence between lexical root word, directional verb and temporal aspect inflections, and the three sign components of location, orientation and handshape as shown in Figure 4.10. From the figure, we note that the handshape phone value depends only on the lexical word value, and thus signs formed from the same lexical root word regardless of their inflectional values, share the same handshape phone sequence. There are 29 lexical root words (see Table B.1), and we additionally define two words, REST_START and REST_END, that represent the signer's hand at rest, at the start and end of a sentence, respectively[2]. Therefore there are 31 possible values for the $Q_t^{1\text{-}LW}$ node and 31 distinct sign models in the handshape

[2] To facilitate manual segmentation of the data into individual sentences, in these experiments the sentences are signed with a pause in between each sentence, during which the signer's hand returns to a rest position. Since it was difficult to determine exactly when the hand has started signing the first sign after moving from the rest position, we considered the sentence to start and end with the signer's hand at rest.

component.

The orientation phone node in Figure 4.10 has as parents only the lexical word and the directional verb inflection nodes. Thus signs formed from the same lexical root word and directional verb inflection combination, regardless of their temporal aspect inflection values, share the same orientation phone sequence. There are 11 possible values for $Q_t^{1\text{-}DV}$ node (see Table B.3). Not all combinations of lexical root word and directional verb inflections are possible – in the experimental vocabulary there are 63 such combinations that appear, thus there are 63 distinct sign models in the orientation component.

The location phone node in Figure 4.10 has as parents all three sign-level nodes, i.e. the lexical word, directional verb inflection and temporal aspect inflection nodes. However, we can take advantage of context-specific independence [22] to reduce the number distinct sign models in the location component to 58 (from 98). Context-specific independence refers to the case where not all parents are always relevant in determining the child's distribution. Some of the parents are irrelevant when the other parents of the child take on specific values, i.e. the independence is according to context. In our case, we make the assumption that when temporal aspect inflection is absent, the lexical word value is not relevant for determining the location component phone. This is graphically represented in Figure 6.2. This is a reasonable assumption since the start and end locations of a sign that has a directional verb inflection depends more on the identity of the subject and object of the verb rather than the lexical word value.

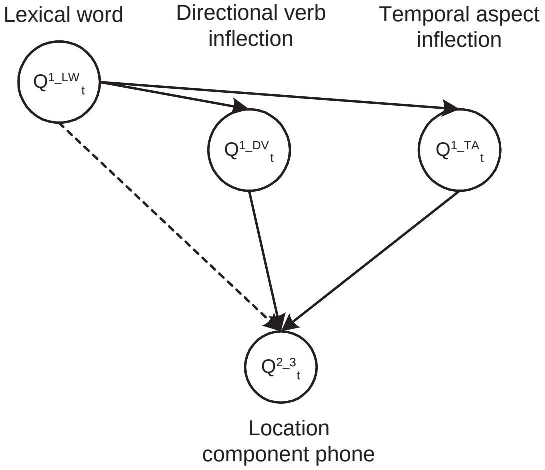


Figure 6.2: Context-specific independence in the causal relationship between lexical word, directional verb inflections, temporal aspect inflections and the location component phone. The causal link in dotted line is absent when there is no temporal aspect inflections, i.e. $Q_t^{1\_TA}$ takes on value of 0.

Section 6.6 describes experiments for training to obtain the final 31, 63 and 58 sign models of the handshape, orientation and location components, respectively.

6.3 Approaches to deal with movement epenthesis

In HMM-based systems for SL recognition, there are three main approaches for dealing with movement epenthesis: modelling signs with context-independent HMMs, modelling signs with context-dependent HMMs, and explictly modelling movement epenthesis. The approach of modelling signs with context-independent HMMs [12, 143], uses one HMM to model each sign (or subunit, in the case of [12]). The same HMM model is used for each sign, regardless of the preceding and following sign in the sentence, i.e. the HMM model is context-independent. The approach of modelling signs with context-dependent biphone HMMs defines a unique HMM for every distinct combination of two signs in sequence. Other works accounted for movement epenthesis by explicitly modeling it. In Assan and Grobel [7] all transitions between signs go through a single state, while in Gao et al. [47] separate HMMs model the transitions between each unique pair of signs that occur in sequence. In more recent experiments [45], the number of such transition HMMs was reduced by clustering the transition frames. In Vogler [157], separate HMMs model the transitions between each unique ending and starting location of signs, and also between each unique ending and starting handshapes of signs. [157] also assessed the advantage of explicit epenthesis modeling by making experimental comparisons with context-independent HMMs (as used in [12, 143]), and context-dependent biphone HMMs. On a test set of 97 sentences constructed from a 53-sign vocabulary, explicit epenthesis modeling was shown to have the best word recognition accuracy (92.1%) while context-independent modeling had the worst (87.7% vs 89.9% for biphone models).

In our experiments we explicitly model movement epenthesis, with the assumption that within each sign component, movement epenthesis appears as a smooth transition between the ending phone of the preceding sign and the starting phone of the following sign. Thus each unique pair of phone values gives a possible movement epenthesis. This would mean 28 x 28 (= 784) possible movement epenthesis in the location component, for example. We reduce this number to 28 (i.e. the same as the number of location phones defined in Section 6.2, based on a reasonable guess that the number of movement epenthesis "phones" should be in the order of the number of phones in signs) by clustering pairs of the 28 cluster centres found in Section 6.2, i.e. for the location component, we cluster 784 6-dimensional vectors to obtain 28 clusters. With this definition of movement epenthesis, in Section 6.6 we train a (single channel) H-HMM (as in Figure 4.8) for the location component, where we define a total of 56 possible values for the phone node, $Q_t^{2_c}$, consisting of 28 values for phones in signs and 28 values for movement epenthesis.

We also experimented with a different approach, which bears some resemblence to the context-independent HMM approach mentioned above. Our approach seeks to extract only data points that correspond to significant points within a sign, for further processing. We conjecture that these significant points within a sign correspond to points where the hand motion exhibit sharp changes in motion direction, and thus use motion direction change in the 3-dimensional hand position trajectory as a criterion for detecting the points. The remaining data points are discarded. As a by-product, this process also removes data points corresponding to movement epenthesis. Since a straight line is the shortest distance between two points in 3-dimensional space, movement epenthesis most often appears as a straight line motion between two points, which are the ending position of the preceding sign and the starting position of the following sign. Straight line motions do not exhibit any sharp movement changes within the period of the motion itself, thus by our procedure, the corresponding data points are not extracted for further processing and are discarded.

The motion direction change detection procedure is described below. We first performed smoothing and interpolation by spline-fitting the 3-dimensional position trajectory of a sentence to 10 times the original number of data points. This process yields equispaced 3-dimensional points between each pair of the original data points but does not produce equi-spaced points across the entire sentence trajectory, since spline-fitted points in slow-moving sections would be closer together. To calculate motion direction change, the motion trajectory shape should not include speed information, i.e. it should be invariant to the signing speed. Hence, we re-interpolate the spline-fitted points to obtain equi-spaced points across the entire sentence trajectory. These are the smoothed points we used for motion direction change detection. Experiments using curvature as a criterion for detecting motion direction change produced very noisy results, and hence we used the change in motion vector angle in successive smoothed points as the detection criterion (see θ_t in Figure 3.4 for an illustration). Changes above a threshold were taken as indicating points with a sharp change in motion direction. These points were then mapped to the closest original data points in the trajectory to mark the points with sharp change in motion direction. Figure 6.3 plots the 3-dimensional trajectory of an example sentence, with the extracted data points shown.

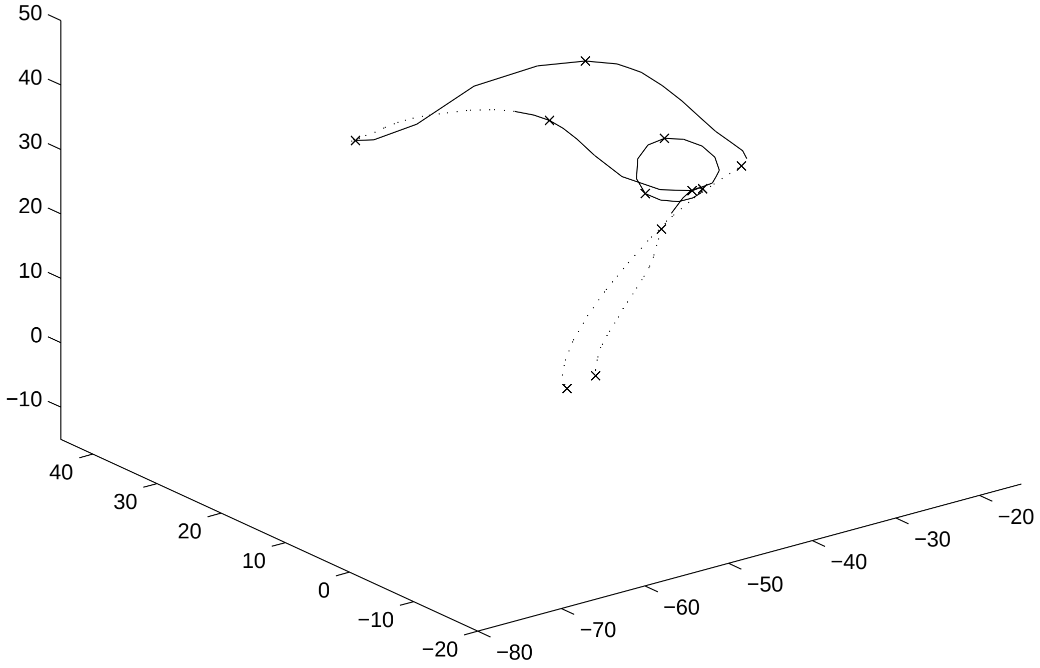


Figure 6.3: Plot of 3-dimensional position trajectory and extracted data points (crosses), for the sentence: GIVE$^{\text{I}\rightarrow\text{YOU}}$ PAPER. Sections of the trajectory corresponding to movement epenthesis is plotted with dotted line, sections of the trajectory corresponding to signs is plotted with solid line.

With this approach, data points corresponding to signs are discarded along with those corresponding to movement epenthesis. This results in each phone within a sign encompassing a much smaller variation in appearance of the observation feature, as compared to when all the original data points are used. In effect, each phone can be represented with just one subphone state and we can use a simpler

H-HMM model with two Q-levels instead of three for training. The two Q-level H-HMM is as shown in Figure 6.4 where the subphone-level nodes and their links have been removed from the H-HMM of Figure 4.8. In the two Q-level H-HMM, since there are no subphone-level states, the phone model simplifies to just the output probability distributions of the component feature, i.e. the CPD of node O_t^c (see Section 4.2.1). Modelling the location component requires a total of 28 possible values for the phone node Q_t^{2c} since movement epenthesis is not modelled. In Section 6.6 we train a two Q-level H-HMM for the location component and compare the test results with the three Q-level H-HMM mentioned above.

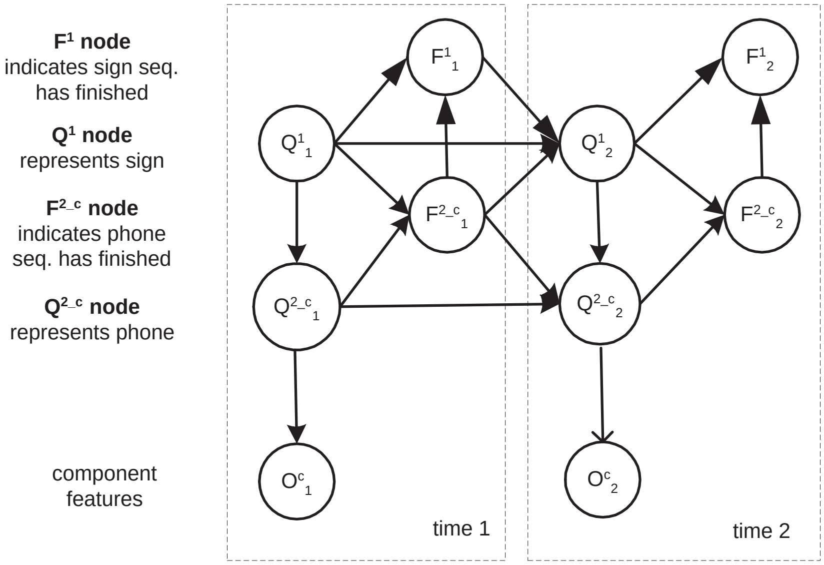


Figure 6.4: H-HMM with two Q-levels for training sign component c. Nodes indexed by superscript c pertain to the specific component (e.g. $Q_t^{2\text{-}c}$ refers to the phone node at time t for component c). Dotted lines enclose nodes of the same time slice.

6.4 Labelling of sign values for subset of training sentences

Our training and modelling strategy involves learning the component-specific sign and phone models by training each component's models independently of each other and with independent observation feature sets (see Section 4.4.2). A drawback of this approach however is that since the different sign components are trained separately, they are implicitly trained with different sign alignments. The corresponding issue was pointed out for the case of multistream data modelling using flat models by Bengio in [17]. As mentioned in Section 4.4.2, we know the correct sign sequence for each training sentence s, and provide this information during training by setting the CPD parameters of the sign-level nodes, Q_t^1 and F_t^1 in the H-HMM of Figure 4.8, to values that only allow the sign sequence of training sentence s to be constructed. However, the sign node Q_t^1 is not observed (its value is not known) since we do not know the correct sign alignment. For each training sequence s, the EM training algorithm (Algorithm 4.1) explores all possible sign alignments subject to the constraint of adhering to the known sign sequence. The posterior distribution terms calculated in the E-step reflect the different weightages given to the possible sign alignments. Since the component-specific H-HMMs are trained separately, there is no requirement for the training process in each component to give the same weightages to these sign alignments. So in a sense, the sign alignments used in different components to learn their respective CPD parameters do not match (across components).

We experimented with alleviating this problem by using labelled sign nodes for the subset of training data that had been manually segmented in time as described in Section 6.2. For this subset, we know the sign alignment within time slices and can thus label the sign node Q_t^1 for the appropriate time slices when training the (single channel) H-HMM across the different sign components. Section 6.6 presents comparative test results between location component H-HMMs trained with a labelled sign nodes on a subset of training data, and training with no such labels.

6.5 Evaluation criteria for test results

A sign is recognized as correct if values of all the sign-level nodes are inferred correctly, i.e. the lexical word, directional verb inflection and temporal aspect inflection values must all be correct. With this criterion, sign accuracy is defined as follows. Let N_s denote the total number of signs appearing in the test set, S_s the number of substitutions, D_s the number of deletions, and I_s the number of insertions. The sign accuracy, Acc_s, is thus:

$$Acc_s \quad = \quad \frac{N_s - S_s - D_s - I_s}{N_s}$$

Sentence accuracy, $AccSent_s$, is defined by the fraction of sentences without any recognition errors.

In Section 6.7 and 6.8 we also calculate the word accuracy. Since the lexical word is factorized as a separate node, we can find the recognition accuracy for just this node, in effect considering different signs with the same lexical word value as equivalent. With N_w denoting the total number of signs appearing in the set, S_w the number of substitutions, D_w the number of deletions, and I_w the number of insertions, the word accuracy, Acc_w, is defined as

$$Acc_w = \frac{N_w - S_w - D_w - I_w}{N_w}$$

Sentence accuracy (when different signs with the same lexical word value are considered as equivalent), $AccSent_w$, is defined as the fraction of sentences without any recognition errors.

6.6 Training and testing on a single component

In the first set of experiments we compared different approaches to dealing with movement epenthesis (see Section 6.3) and also examined the possible benefits of using labelled sign nodes for a subset of training data (see Section 6.4). We experimented with training to obtain three different trained H-HMMs for the location component. In all three models the observation features are $\underline{o}_H$, as described in

Section 6.1.2. The training procedure is as described in Section 4.4.2. Starting from initial model parameters for the H-HMM, the iterative steps in the EM algorithm are repeated until it converged. Training uses constrained sentence models reflecting the correct sign sequence in training sentences. In the E-step, inferencing uses the forward interface inferencing algorithm for DBNs [110][3].

The first model trained is a H-HMM with three Q-levels (Figure 4.8), modelling movement epenthesis explicitly, as described in Section 6.3. The phone sequences obtained from the winning cluster sequence (for the location features) of each sign are used to initialize the sign model parameters, i.e. the CPD parameters for the $Q_t^{2\text{-}c}$ and $F_t^{2\text{-}c}$ nodes (see Section 6.2). We define three subphone states for each of the phone models, with state transitions defined as the 3-state left-right (Bakis) model (see Figure 4.2). The CPD parameters for the $Q_t^{3\text{-}c}$ and $F_t^{3\text{-}c}$ nodes are initialized such that the nonzero subphone state initial, transition and ending probabilities are equiprobable. For each phone, the mean and covariance of the three subphone output probability distributions are initialized identically, to the corresponding cluster's mean and covariance values (see Section 6.2). This defines the initial CPD parameters of O_t^c. During training, the observation features for the H-HMM are obtained from every time slice of the training sequences.

[3] All experiments in this chapter were performed using Matlab code based on the Bayes Net Toolbox [108].

The second model trained is a H-HMM with two Q-levels (Figure 6.4), as described in Section 6.3. As in the three Q-level H-HMM above, the phone sequences obtained from the winning cluster sequence (for the location features) of each sign are used to initialize the sign model parameters, i.e. the CPD parameters for the $Q_t^{2\text{-}c}$ and $F_t^{2\text{-}c}$ nodes. In this two Q-level H-HMM, there are no subphones, hence there is only output probability distribution for each phone and its mean and covariance is initialized to the corresponding cluster's mean and covariance values (see Section 6.2). This defines the initial CPD parameters of O_t^c. The observation features for the model are obtained from data points extracted using the motion direction change detection procedure described in Section 6.3.

The third model trained is a H-HMM with two Q-levels with model parameters initialized exactly as above and with the same observation features. The sole difference is that during training, a subset of the training sequences have observed (known) sign node values. These are the sentences that were manually segmented to obtain initial model parameters as described in Section 6.2. Thus the sign node values for these sentences are already known.

The three trained models above are tested for sign recognition on the test sentence set. Inferencing during testing obtains the most-probable assignment of values to all the hidden nodes in the model (see Section 5.1). We use the forward interface algorithm in this decoding step. The sign and sentence accuracy results for the three trained models are shown in Table 6.1.

Table 6.1: Test results on location component H-HMMs.

Trained model	Acc_s (%)	$AccSent_s$ (%)	D_s	S_s	I_s	N_s
3 Q-level H-HMM with movement epenthesis modelling	69.7	15.5	26	185	28	788
2 Q-level H-HMM without movement epenthesis modelling	78.3	18.3	11	148	12	788
2 Q-level H-HMM with labelled sign nodes	78.4	18.3	11	150	9	788

The best accuracy results were obtained with the third model, the two Q-level H-HMM, without modelling movement epenthesis, and presented with training sequences where a subset was labelled with sign node values. The results of the third model was only marginally better than that of the second model (which did not use sign labels). However, since there is no added training effort required to obtain the sign labels (the labels were obtained as a by-product of the parameter initialization procedure described in Section 6.2), the third trained model will be used in the next section to provide the necessary location-specific CPD parameters to construct the MH-HMM. We then applied the training strategy employed for the third model above, to train two Q-level H-HMMs for the handshape and orientation components. The observations features for these two components are as described in Section 6.1.2. The sign and sentence accuracy results for the trained models of handshape and orientation components are shown in Table 6.2. The accuracy results reported in Table 6.1 and 6.2 are quite low. This is not unexpected

Table 6.2: Test results on trained models for two Q-level H-HMMs for handshape and orientation components.

Trained model	Acc_s (%)	$AccSent_s$ (%)	D_s	S_s	I_s	N_s
Handshape component H-HMM	73.1	12.7	11	199	2	788
Orientation component H-HMM	85.0	36.6	16	95	7	788

since there are 98 signs in the vocabulary, but fewer than 98 distinct sign models in each of the components - there are 58 distinct sign models defined for the location component, 31 for handshape component and 63 for orientation component. Thus not all of the 98 signs can be distinguished based on any of the components singly. The motivation for testing single component trained models is to make sure that sign/sentence accuracy results are in a reasonable range, before proceeding to construct the MH-HMM based on the parameters of these trained models.

Examination of the test results on location component H-HMMs shows that the accuracy results with the three Q-level H-HMM is quite low in comparison to that obtained with the simpler two Q-level H-HMM. The accuracy of the three Q-level H-HMM could be affected by the fact that the “phones” corresponding to movement epenthesis are clustered versions of pairs of location phones corresponding to signs. That is to say, not every unique pair was defined as a unique movement epenthesis. This was necessary to reduce the number of movement epenthesis to be modelled but may have resulted in a loss of modelling accuracy. The large number of movement epenthesis models appears to be a problem in the approach of modelling movement epenthesis explicitly, and often there are more movement epenthesis models than there are phone models corresponding to signs. For example [157] defined 78 phone models corresponding to signs and 133 movement epenthesis models (i.e. there were 70% more movement epenthesis models than phone models). This seems like a waste of training data and resources. Since there are more movement epenthesis models than there are phone models, the majority of training data is used to learn parameters of movement epenthesis models!

Our approach is a viable alternative, discarding data points corresponding to movement epenthesis. Training data is used to learn just the phone models instead of models of the transitions between signs. Training time is reduced dramatically because there are much fewer data points.

6.7 Testing on combined model

A MH-HMM modelling the location, handshape and orientation components is constructed by combining the component-specific sign and phone models trained in Section 6.6 (also see Section 4.4.1). The MH-HMM is shown in Figure 6.5. We presented the observed values of the component features O_t^c, for components $c = 1, 2, 3$ from the test set sentences. Synchronization between component streams at sign boundaries was enforced by setting $S_t^2 = 1$, for $1 \leq t \leq T$. We also set $F_t^1 = 0$ for $t = 1 \ldots, T-1$ and as $F_T^1 = 1$, indicating that for each test sequence, the

Table 6.3: Test results on MH-HMM combining trained models of location, handshape and orientation components.

Num. of samples	Acc_s (%)	$AccSent_s$ (%)	D_s	S_s	I_s	N_s	Acc_w (%)	$AccSent_w$ (%)	D_w	S_w	I_w	N_w
3000	92.0	58.5	7	56	0	788	98.4	92.3	7	8	0	788
5000	92.4	62.0	4	53	3	788	98.9	95.1	4	2	3	788
10000	92.6	61.3	6	50	2	788	98.5	92.3	6	4	2	788
15000	92.6	62.7	5	50	3	788	98.7	94.4	5	2	3	788
20000	93.4	66.2	4	45	3	788	98.9	95.1	4	2	3	788
25000	93.9	68.3	5	42	1	788	98.7	93.7	5	4	1	788
30000	92.9	64.8	8	45	3	788	98.4	92.3	8	2	3	788
40000	93.7	68.3	6	40	4	788	98.4	92.3	6	3	4	788

sentence ends only at the last time slice and not before. With these observed node values, the most probable sign sequence in each sentence was inferred using particle filtering (PF) as described in Chapter 5 and in particular Section 5.5. The sign and word accuracy results for this MH-HMM are shown in Table 6.3 for different number of samples used in the PF algorithm. Only one trial was performed at each sampling level.

The sign recognition accuracy is greatly improved compared to single component decoding results (compare Tables 6.1 and 6.2). Within each of the components, there are less than 98 distinct (component-specific) sign models. However, although multiple signs may share the same component-specific sign model, none of the signs share the same component-specific sign models in all three components. That is to say that the 98 signs in the vocabulary have distinct combinations of

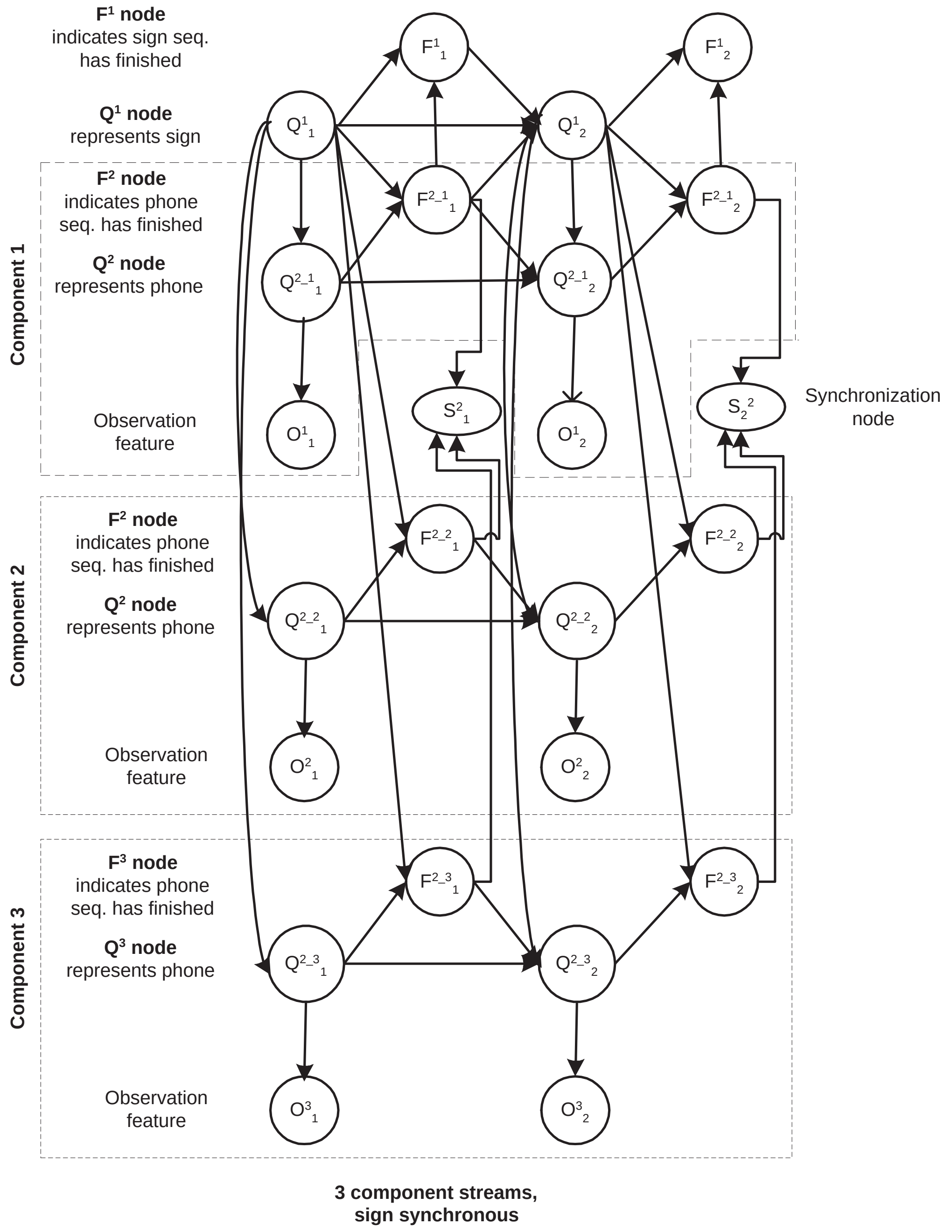


Figure 6.5: MH-HMM with two Q-levels and with synchronization between components at sign boundaries (shown for a model with three components streams, and two time slices). Dotted lines enclose component-specific nodes.

component-specific sign model. Thus the improved sign recognition results is to be expected.

The PF algorithm is expected to give better inferencing results with increased number of samples, theoretically approaching results that would be obtained using exact inferencing at the limit of infinite number of samples. The results in Table 6.3 show an improvement in sentence accuracy, $AccSent_s$, with increased number of samples. It might be worth increasing the number of samples beyond the maximum 40000 that we experimented with, to investigate if this would produce further improvement in the sentence accuracy, which is currently quite low considering the relatively high sign accuracy.

The correspondence between sample number and accuracy is however not seen in the other accuracy measurements. The maximum sign accuracy (Acc_s) of 93.9% was obtained with 25000 samples and not with the maximum number of 40000 samples that we ran experiments with. The maximum word accuracy (Acc_w) of 98.9% was also obtained with fewer than 40000 samples. Due to the stochastic nature of the inferencing algorithm, multiple trials at each sampling level are required before we can conclude if there is indeed diminishing returns in sign and word accuracy beyond 20000 to 25000 samples.

The majority of the errors made in sign recognition are substitution errors, this, together with the much improved accuracy results, Acc_w and $AccSent_w$, when we only infer the lexical word value, indicate that many of the errors made in sign recognition involved errors in determining the inflection values and not the lexical word values. Although we did not calculate the recognition confusion matrix, the above results would indicate that a large proportion of signs that get confused are signs that appear in various inflected and non-inflected versions in the vocabulary. In general, the frequency of a sign appearing in the training set (and thus the amount of training data available for that sign) would effect recognition accuracy for that sign. Thus recognition rates for sparsely-represented signs may suffer. It is reasonable however to also surmise that signs based on the same lexical word are inherently more difficult to distinguish from one another since they share common features – for example the handshape. The results above seem to bear this out. Acc_w and $AccSent_w$ seem to saturate quickly with the number of samples, for example, the best word accuracy results are found with 5000 samples and 20000 samples.

6.8 Testing on combined model with training on reduced vocabulary

In this set of experiments, we applied the same strategy as outlined in Section 6.6 for training H-HMMs for each of the sign component but withheld a subset of the sentences from the training set. Specifically, sentences containing 16 out of the 98 signs in the vocabulary were not presented to the models during training and were not used in obtaining winning cluster sequences to initialize the sign model parameters (see Table B.4 for the set of unseen signs). Thus the three components models were trained on 144 sentences containing 835 signs, instead of the full training set of 201 sentences containing 1139 signs. Despite not seeing all the signs in the vocabulary, it was still possible to train the full set of component-specific sign models because multiple signs share the same component-specific models. This is due to the structural conditional independencies and the context-specific independencies in the models (see Section 6.2). The set of excluded signs was chosen with the requirement that for all three components, each distinct component-specific sign model must be represented among the remaining signs. As adequate training data is required for robust learning of parameters, another requirement is that at least 5 sentences containing signs that share the same component-specific sign model must be present among the sentences used for training.

Once the models were trained, we constructed an MH-HMM by combining the component-specific sign and phone models of location, handshape and orientation. As in the previous section, we presented the observation features for all three components of the test set sentences and set the values of the S_t^2 and F_t^1 nodes. The most probable sign sequence in each sentence was inferred as before.

Within the test set, some of the sentences contained only signs that had been

Table 6.4: Test results on MH-HMM combining trained models of location, handshape and orientation components, tested on sentences with only seen signs.

Num. of samples	Acc_s (%)	$AccSent_s$ (%)	D_s	S_s	I_s	N_s	Acc_w (%)	$AccSent_w$ (%)	D_w	S_w	I_w	N_w
3000	92.2	62.2	6	34	4	563	98.2	89.8	6	0	4	563
5000	92.9	65.3	6	34	0	563	98.2	90.8	6	4	0	563
10000	93.6	68.4	7	29	0	563	98.1	89.8	7	4	0	563
15000	95.2	77.6	5	22	0	563	98.4	91.8	5	4	0	563
20000	95.0	74.5	4	23	1	563	98.2	91.8	4	5	1	563
25000	95.2	76.5	4	21	2	563	98.4	93.9	4	3	2	563
30000	95.9	78.6	3	17	3	563	98.9	93.9	3	0	2	563
40000	94.1	73.5	8	23	2	563	97.9	89.8	8	2	2	563

present during training, i.e. seen signs, while others contained signs that had not been present during training, i.e. unseen signs. The sign and word accuracy results for each case are shown in Table 6.4 and 6.5, for different number of samples used in the PF algorithm.

Sign recognition accuracy from decoding sentences that only contain seen signs are better than those obtained when we use the full set of training sentences that contained all 98 signs (compare Table 6.3). This makes sense because in the former case we trained component-specific sign models on a smaller training set with less variations in sign appearances, and then tested on representative sentences containing the same signs.

Accuracy results from decoding sentences that contained unseen signs were not very good. But the word accuracy results, Acc_w and $AccSent_w$, for these sentences

Table 6.5: Test results on MH-HMM combining trained models of location, handshape and orientation components, tested on sentences containing unseen signs.

Num. of samples	Acc_s (%)	$AccSent_s$ (%)	D_s	S_s	I_s	N_s	Acc_w (%)	$AccSent_w$ (%)	D_w	S_w	I_w	N_w
3000	84.4	27.3	2	32	1	225	98.7	93.2	2	0	1	225
5000	85.8	31.8	3	29	0	225	98.7	93.2	3	0	0	225
10000	86.7	34.1	2	28	0	225	99.1	95.5	2	0	0	225
15000	88.0	40.9	1	26	0	225	99.6	97.7	1	0	0	225
20000	86.7	36.4	2	28	0	225	99.1	95.5	2	0	0	225
25000	87.6	38.6	1	27	0	225	99.6	97.7	1	0	0	225
30000	88.4	43.2	1	25	0	225	99.6	97.7	1	0	0	225
40000	88.0	40.9	1	26	0	225	99.6	97.7	1	0	0	225

showed a significant increase (even compared to the word accuracy results from decoding sentences that only contain seen signs). This makes sense since each lexical word in the vocabulary is represented in the training set, so for each of the unseen signs, the uninflected version and possibly other inflected versions based on the same lexical word were seen in the training set.

In conclusion, results on recognizing continuous signs by combining information from multiple sign components using the MH-HMM are promising (Table 6.3), obtaining a maximum sign accuracy of 93.9%. The test sentences included signs which contained inflection meaning but if we only consider recognition of lexical meaning, the word accuracy improves to a maximum of 98.9%. Results in this section also show that with our approach of defining multiple signs as sharing the same component-specific sign models, it was possible to recognize continuously signed sentences containing unseen signs, albeit at a lower recognition accuracy – the best result obtained was 88.4% (Table 6.5). The lexical meaning of these signs were recognized at a much higher accuracy of 99.6%, indicating that most of the recognition errors were made in inferring the inflection values.

Chapter 7

Conclusions and future work

7.1 Contributions

The main contribution of this thesis is in addressing an aspect of SL that has largely been overlooked in previous work on SL recognition and yet is integral to signed communication. The work described in thesis is the most comprehensive to-date on the recognition of the complex variations in sign appearances due to grammatical processes. These processes systematically change both the temporal and spatial dimensions of a root sign word to convey information in addition to lexical meaning. The systematic modulations in sign appearance that are recognized in this work are of a nature and number that have not been tackled in previous work. Furthermore, we also extracted information conveyed through multiple simultaneous modulations on sign appearance, which is likewise a novel contribution.

We presented the MH-HMM as a modelling and recognition framework for continuously signed sentences that include modulated signs. The MH-HMM models the hierarchical, sequential and parallel organization in signing while requiring synchronization between parallel data streams at sign boundaries. The effect of grammatical processes on sign appearance is learned from data in a modular way; this simplifies training while still being able to model the complex effects of these processes. In this thesis we showed how the MH-HMM can be applied to our problem domain, and described how the PF algorithm can be specifically applied in our model to infer the most-likely sign sequences in continuous sentences.

We propose the MH-HMM not only as a model suitable for the problem domain that is the focus of this thesis but for any domain where there is a hierarchy of abstract levels, multiple time scales, and multiple data streams which require synchronization between the streams. Previous work in domains exhibiting hierarchical and parallel structure either separated the levels of hierarchy (layered HMMs) or had a parallel flat structure (PaHMM, product HMM). The MH-HMM models both hierarchical and parallel structure, while retaining modularity. MH-HMM has advantages over the existing methods mentioned above, including:

- Hierarchical and parallel structures in the data are modelled simultaneously, allowing information at all levels to influence the final inferencing results. The information flow is not exclusively top-down or bottom-up (as in layered HMMs).

- Factorization of the system state allows information to be input where available, i.e. a subset of the nodes can be labelled during testing.
- The parameters of the model pertaining to different data streams are learned separately, making the training faster and easier.
- The framework is modular and flexible; we could, for example, combine H-HMMs with different number of Q-levels together into the MH-HMM. We can easily experiment with enforcing synchronization between streams at different levels, for example in speech, at the word or phone level.

The MH-HMM is a probabilistic model and all the parameters are learned from data, including the probabilistic relationship between lexical and grammatical information conveyed in signs and sign subunits which are the equivalent of phones in speech. This is different from previous work which defined the relationship between lexical information and phones linguistically or was based on a phonological model instead of learning from data. The ability to learn the effect of grammatical processes on sign appearance from data is especially pertinent for SL recognition because unlike in speech, there is no consensus on a phonological model for signs and thus there is no equivalent to the pronunciation dictionaries as used in speech recognition to define words as a decomposition of phones.

In another important contribution of our work we showed how to take advantage of commonalities between how grammatical processes affect appearances of different root sign words to reduce parameters learned in the model and recognize new and unseen combinations of root words and grammatical information. This is crucial because there is a large variety of information that can be conveyed in addition to the lexical meaning in signs and hence a large variety of appearance changes that can occur to a root word, making it impossible to obtain training data for all these appearances.

Our work also proposed a novel method of dealing with sign transitions (movement epenthesis) in the data stream. Our method performs better than the approach of explicitly modelling movement epenthesis and circumvents the problem that arises in the latter approach whereby the majority of the training resources ends up being used for training sign transitions rather than actual signs.

7.2 Future Work

Future work should include data from both hands in continuous sign recognition. Although the sign vocabulary considered in our experiments included two-handed signs, all the signs were distinguishable by looking only at the dominant hand. However, this is not true in general as we consider larger sign vocabularies. In classifier signs the non-dominant hand is especially important for expressing relative spatial relations. Including data from both hands would require adding additional channels to the MH-HMM model.

In the isolated gesture experiments we had separated out movement attributes affected by grammatical processes as separate sign components. However this was not done in modelling continuous signs and the associated experiments. Modelling movement attributes as a separate data stream is a difficult problem as we require features that evolve as a quasi-stationary process and at the same time are invariant to position. Some features currently being explored include curvature (although it has been found to be noisy), centroid distance function [9] and Fourier transform based features.

The PF algorithm used for inference in the MH-HMM is a sampling method that is relatively easy to implement on different types of DBNs without needing to customize the basic algorithm to the specific model. Its main disadvantage however is that since a large number of samples are required to represent the distribution of a large state space, it runs very slowly for large models. At each time step, the number of operations required for generating samples is $O(SK)$, and for weighting the samples is $O(SL)$ (where S = number of samples, K = total number of hidden variables, L = total number of observation variables. See Section 5.4.). For example, in Section 6.7 testing was on the combined model of Figure 6.5 which has 7 hidden variables (K) and 5 observed variables (L). Thus the number of operations per time step was in the order of 36,000 ($SK+SL$) when 3000 particles were used, and in the order of 480,000 when 40,000 particles were used. One possible avenue for exploration is the use of Rao-Blackwellised particle filtering (RBPF) which combines exact and stochastic inferencing, resulting in a smaller state space that requires fewer samples for representation. Both PF and RBPF are stochastic in nature, thus ideally multiple trials at each sample level should be performed in experiments. This we did not do in the experiments of Chapter 6 and should be looked into in future work. It would also be informative to explore the use of non-stochastic approximate inference methods such as loopy belief propagation [124, 113] and variational methods [71].

The focus of the experiments in Chapter 6 was in verifying the feasibility of using the MH-HMM to recognize inflected signs. As such it is a reasonable first step to perform the experiments with only a single signer. We would need to include more signers in the future to see how recognition results would be affected when there are multiple signers, especially in light of the findings in Chapter 3 that there is indeed much variation in how different people perform the same gestures. To reduce signer variation in feature measurements, data collected from the Virtual Technologies Cyberglove should be calibrated carefully (a calibration software application is provided by the glove manufacturer). In addition, we could calibrate position information measured from the Polhemus electromagnetic tracker by scaling position measurements according to the extent of each signer's arm reach.

The signer used in the experiments of Chapter 6 is a native signer but not a native ASL signer, whereas the sentences signed in the experiments are ASL sentences. To apply our model to a native ASL signer, we would need to retrain the model, possibly including redefining the phones of each component according to the data collected from the new ASL signer. However, the structure of the model, i.e. the MH-HMM formulation, should remain the same, as this was derived based on the structure of sign sentences in general, and specifically ASL sentences and grammar.

One of the considerations in designing a DBN is the number of hidden variables and the number of variable states. Inappropriate numbers could lead to under-fitting or over-fitting. In our design, the hidden variables of the MH-HMM represent the sign, phones of different components and HMM states of the phone models of different components. Thus the number of hidden variables is determined by the hierarchical and parallel nature of the domain data. The number of states of the phone variable in each component was determined by clustering the data (see Section 6.2), thus it is also data-driven and reflects variation present in the data. The HMM phone models were all designed with 3 states. Although this number was just a reasonable guess, it does not affect the eventual test results reported in Sections 6.7 and 6.8 where tests were performed on the two Q-level MH-HMM which models the phone level but not the HMM state level.

The proposed MH-HMM is a generative model and thus shares some of the same disadvantages as simpler generative DBNs such as HMMs. This includes difficulty in incorporating long-range dependencies between the states and the observations and the requirement of conditional independence of observations (from different time frames). Discriminative models such as Conditional Random Fields

(CRF) [89] and Hidden Conditional Random Fields (HCRF) [130] are able to model sequence data without having the disadvantages mentioned above. CRFs have been used successfully in parts-of-speech tagging [140], information extraction [101, 139, 31], RNA structural alignment [138], protein structure prediction [96], labeling and segmenting images [58, 88], to name a few. However the CRF is not multi-layered and does not explicitly model intermediate structures in the manner of the H-HMM. HCRFs are multi-layered models with hidden states and have been used for recognizing isolated gestures [130] and classifying segmented phones [55]. However, HCRF requires training data where the top-level variable (i.e. the variable to be inferred) is labelled for all time frames. In our present application, the top-level variable is the sign value. The majority of our training sentences do not have labelled sign values, and thus could not be used for training a model such as the HCRF. Another disadvantage of discriminative models is that all model parameters would need to be re-learned if the model is to be expanded to include new sign vocabulary. A generative model such as the MH-HMM would only need to learn parameters pertaining to the new vocabulary, keeping intact the parameters that have already been learned for existing vocabulary.

There are many problem domains that require modelling of multiple observation streams corresponding to the same sequence of events and subsequent recognition of these events. The continuous events to be recognized include multiband speech, audio-visual speech, gesture, human activity, group action in meetings, and facial expressions. Many of these events can also be decomposed into a hierarchical structure. MH-HMM can be applied to model these sequential events if they can be analysed such that there is a definable set of “words” in the vocabulary, the equivalent of “phones” or subunits in each observation stream, and with synchronization between streams at either the word or phone level. It would be of great interest to apply the MH-HMM to such problems and compare the results to existing methods and architecture for modelling in these domains.

Bibliography

[1] *DataGlove Model 2 User's Manual.* VPL Research Inc, Redwood City, CA, 1987.

[2] *Polhemus 3Space User's Manual.* Polhemus, Colchester, VT, 1991.

[3] *Power Glove Serial Interface (2.0 Ed.).* Student Chapter of the ACM, UIUC, 1994.

[4] *CyberGlove User's Manual.* Virtual Technologies Inc., 1995.

[5] S. Akyol and U. Canzler. An information terminal using vision based sign language recognition. In *Proc. ITEA Workshop Virtual Home Environments*, volume C-LAB, pages 61–68, 2002.

[6] O. Al-Jarrah and A. Halawani. Recognition of gestures in Arabic Sign Language using neuro-fuzzy systems. *Artifi. Intell.*, 133(ER1-2):117–138, Dec 2001.

[7] M. Assan and K. Grobel. Video-based sign language recognition using hidden Markov models. In *Proc. Gesture Workshop*, pages 97–109, 1997.

[8] C. Baker-Shenk and D. Cokely. *American Sign Language: A Teacher's Resource Text on Grammar and Culture.* Clerc Books, Gallaudet Univ. Press, Washington, D.C., 1981.

[9] F. Bashir, A. Khokhar, and D. Schonfeld. A hybrid system for affine-invariant trajectory retrieval. In *ACM SIGNUM Multimedia Information Retrieval Workshop*, pages 235–242, 2004.

[10] R. Battison. *Lexical Borrowing in American Sign Language.* Linstok Press, Silver Spring, MD, updated and re-issued in 2003 edition.

[11] B. Bauer and K.-F. Kraiss. Towards a 3rd generation mobile telecommunication for deaf people. In *Proc. 10th Aachen Sympos. Signal Theory Algorithms and Software for Mobile Comms.*, pages 101–106, Sep 2001.

[12] B. Bauer and K.-F. Kraiss. Towards an automatic sign language recognition system using subunits. In *Proc. Gesture Workshop*, pages 64–75, 2001.

[13] B. Bauer and K.-F. Kraiss. Video-based sign recognition using self-organizing subunits. In *Proc. Int'l Conf. Pattern Recogn.*, volume 2, pages 434–437, 2002.

[14] U. Bellugi and E. S. Klima. Aspects of sign language and its structure. In J. F. Kavanagh and J. E. Cutting, editors, *The Role of Speech in Language*, pages 171–203, Cambridge , MA, 1975. MIT Press.

[15] S. Bengio. Multimodal speech processing using asynchronous hidden Markov models. *Information Fusion*, pages 81–89, 2004.

[16] S. Bengio and Y. Bengio. An EM algorithm for asynchronous input/output hidden Markov models. In *Proc Int'l Conf. Neural Information Processing, ICONI*, 1996.

[17] S. Bengio and H. Bourlard. Multi channel sequence processing. In J. Winkler, N. Lawrence, and M. Niranjan, editors, *Machine Learning Workshop, LNAI*, volume 3635, pages 22–36, Berlin Heidelberg, 2005. Springer-Verlag.

[18] Y. Bengio. Markovian models for sequential data. *Technical Report 1049, Dept. IRO, Universit'e de Montr'eal*, 1996.

[19] P. Beyerlein. Discriminative model combination. In *Proc. Int'l Conf. on Acoustics, Speech and Signal Processing*, volume 1, pages 481–484, 1998.

[20] H. Birk, T. Moeslund, and C. Madsen. Real-time recognition of hand alphabet gestures using principal component analysis. In *Proc. Scandinavian Conf. Image Analysis*, pages 261–268, 1997.

[21] H. Bourlard and S. Dupont. A new ASR approach based on independent processing and recombination of partial frequency bands. In *Proc. Int'l Conf. on Spoken Language Processing*, volume 1, pages 426–429, 1996.

[22] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in Bayesian networks. In *UAI*, 1996.

[23] X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In *UAI*, 1998.

[24] A. Braffort. Argo: An architecture for sign language recognition and interpretation. In *Proc. Gesture Workshop*, pages 17–30, 1996.

[25] A. Braffort. Research on computer science and sign language: ethical aspects. In *Proc. Gesture Workshop*, pages 1–8, 2001.

[26] M. Brand, N. Oliver, and A. Pentland. Coupled hidden Markov models for complex action recognition. In *Proc. IEEE Int'l Conf. Comp. Vision and Pattern Recogn.*, page 994999, June 1997.

[27] D. Brentari. Sign language phonology: ASL. In J. Goldsmith, editor, *The handbook of phonological theory*, pages 615–639, Oxford, 1995. Blackwell.

[28] S. Chu and T. Huang. Audio-visual speech modeling using coupled hidden Markov models. In *Proc. IEEE Int'l Conf. Acoustics, Speech, Signal Processing*, page 20092012, May 2002.

[29] E. Cox. Adaptive fuzzy systems. *IEEE Spectrum*, pages 27–31, Feb 1993.

[30] Y. Cui and J. Weng. Appearance-based hand sign recognition from intensity image sequences. *Computer Vision Image Understanding*, 78(2):157–176, 2000.

[31] A. Culotta, R. Bekkerman, and A. McCallum. Extracting social networks and contact information from email and the web. In *First Conf. on Email and Anti-Spam (CEAS)*, Mountain View, CA, 2004.

[32] K. Daoudi, D. Fohr, and C. Antoine. Dynamic Bayesian networks for multiband automatic speech recognition. *Computer Speech and Language*, 17(2-3):263–285, Apr-Jul 2003.

[33] F. Dellaert. *The Expectation Maximization Algorithm.* Tech. report GIT-GVU-02-20, College of Computing GVU Center, Georgia Institute of Technology, Feb 2002.

[34] J. J. Deller, J. Hansen, and J. Proakis. *Discrete-Time Processing of Speech Signals.* Wiley-IEEE Press, 1999.

[35] J.-W. Deng and H. Tsui. A novel two-layer PCA/MDA scheme for hand posture recognition. In *Proc. Int'l Conf. Pattern Recogn.*, volume 1, pages 283–286, 2002.

[36] A. Doucet. *On sequential simulation-based methods for* Bayesian filtering. Technical report CUED/F-INGENG/TR 310, Signal Processing Group, Dept. of Engineering, University of Cambridge, 1998.

[37] A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proc. 16th Conf. Uncertainty in Artificial Intell.*, July 2000.

[38] R. Duda, P. Hart, and D. Stork. *Pattern Classification.* Wiley-Interscience, New York, 2nd edition, 2001.

[39] S. Dupont and J. Luettin. Audio-visual speech modeling for continuous speech recognition. *IEEE Trans. Multimedia*, 2(3):141151, 2000.

[40] A. Edwards. Progress in sign language recognition. In *Proc. Gesture Workshop*, pages 13–21, 1997.

[41] R. Erenshteyn, P. Laskov, R. Foulds, L. Messing, and G. Stern. Recognition approach to gesture language understanding. In *Proc. Int'l Conf. Pattern Recogn.*, volume 3, pages 431–435, 1996.

[42] G. Fang, W. Gao, X. Chen, C. Wang, and J. Ma. Signer-independent continuous sign language recognition based on SRN/HMM. In *Proc. Gesture Workshop*, pages 76–85, 2001.

[43] G. Fang, X. Gao, W. Gao, and Y. Chen. A novel approach to automatically extracting basic units from Chinese Sign Language. In *Int'l Conf. Pattern Recognition*, volume 4, pages 454–457, 2004.

[44] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden Markov model: Analysis and applications. *Machine Learning*, 32:41–62, 1998.

[45] W. Gao, G. Fang, D. Zhao, and Y. Chen. Transition movement models for large vocabulary continuous sign language recognition. In *Proc. Int'l Conf. Auto. Face & Gest. Recog.*, pages 553–558, 2004.

[46] W. Gao, J. Ma, S. Shan, X. Chen, W. Zheng, H. Zhang, J. Yan, and J. Wu. Handtalker: A multimodal dialog system using sign language and 3-d virtual human. In *Proc. Int'l Conf. Advances in Multimodal Interfaces*, pages 564–571, 2000.

[47] W. Gao, J. Ma, J. Wu, and C. Wang. Sign language recognition based on HMM/ANN/DP. *Int'l J. Pattern Recogn. Artif. Intell.*, 14(5):587–602, 2000.

[48] A. Garg, V. Pavlovic, and J. Rehg. Boosted learning in dynamic Bayesian networks for multimodal speaker detection. *Proceedings of the IEEE*, 91(9):1355–1369, Sep 2003.

[49] J.-L. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate gaussian mixture observation of Markov chain. *IEEE Trans. Speech and Audio Processin*, 2:291–298, Apr 1994.

[50] Z. Ghahramani. Learning dynamic Bayesian networks. In C. Giles and M. Gori, editors, *Adaptive Processing of Sequences and Data Structures, Lecture Notes in Artificial Intelligence*, pages 168–187, Berlin, 1998. Springer-Verlag.

[51] Z. Ghahramani and M. Jordan. Factorial hidden Markov models. In D. Touretzky, M. Mozer, and M. Hasselmo, editors, *Proc. Conf. Advances in Neural Information Processing Systems*, volume 8, page 472478, Cambridge,Mass, 1995. MIT Press.

[52] H. Glotin and F. Berthommier. Test of several external posterior weighting functions for multiband full combination ASR. In *Proc. Int'l Conf. on Spoken Language Processing*, volume I, page 333336, 2000.

[53] J. Gowdy, A. Subramanya, C. Bartels, and J. Bilmes. DBN based multistream models for audio-visual speech recognition. In *IEEE Intl Conf Acoustics, Speech and Signal Processing*, May 2004.

[54] G. Gravier, G. Potamianos, and C. Neti. Asynchrony modeling for audio-visual speech recognition. In *Proc. Human Language Technology Conf.*, Mar 2002.

[55] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt. Hidden conditional random fields for phone classification. In *Intl. Conf. on Speech Communication and Technology.* International Speech Communication Association, 2005.

[56] L. Gupta and S. Ma. Gesture-based interaction and communication: Automated classification of hand gesture contours. *IEEE Trans. Syst., Man, Cybern., Part C: Applicat. Rev.*, 31(1):114–120, Feb 2001.

[57] M. Handouyahia, D. Ziou, and S. Wang. Sign language recognition using moment-based size functions. In *Proc. Int'l Conf. Vision Interface*, pages 210–216, 1999.

[58] X. He, R. S. Zemel, and M. A. Carreira-Perpinian. Multiscale conditional random fields for image labelling. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 2004.

[59] D. Heckerman. *A Tutorial on Learning with* Bayesian Networks. Technical Report, Microsoft Research, Mar 1995.

[60] H. Hermansky, S. Tibrewala, and M. Pavel. Towards ASR on partially corrupted speech. In *Proc. Int'l Conf. on Spoken Language Processing*, page 462465, 1996.

[61] J. Hernandez-Rebollar, N. Kyriakopoulos, and R. Lindeman. A new instrumented approach for translating American Sign Language into sound and text. In *Proc. Int'l Conf. Auto. Face & Gest. Recog.*, pages 547–552, 2004.

[62] J. Hernandez-Rebollar, R. Lindeman, and N. Kyriakopoulos. A multi-class pattern recognition system for practical finger spelling translation. In *Proc. Int'l Conf. Multimodal Interfaces*, pages 185–190, 2002.

[63] H. Hienz, K. Grobel, and G. Offner. Real-time hand-arm motion analysis using a single video camera. In *Proc. Int'l Conf. Auto. Face & Gest. Recog.*, pages 323–327, 1996.

[64] E.-J. Holden and R. Owen. Visual sign language recognition. In *Proc. Int'l Workshop on Theoretical Foundations of Computer Vision*, pages 270–287, 2000.

[65] G. Hommel, F. Hofmann, and J. Henz. The TU berlin high-precision sensor glove. In *Proc the WWDU'94 4th Int'l Scientific Conf.*, volume 2, pages F47–F49, 1994.

[66] C.-L. Huang and W.-Y. Huang. Sign language recognition using model-based tracking and a 3D Hopfield neural network. *Machine Vision and Applicat.*, 10:292–307, 1998.

[67] K. Imagawa, S. Lu, and S. Igi. Color-based hand tracking system for sign language recognition. In *Proc. Int'l Conf. Auto. Face & Gest. Recog.*, pages 462–467, 1998.

[68] K. Imagawa, H. Matsuo, R. Taniguchi, D. Arita, S. Lu, and S. Igi. Recognition of local features for camera-based sign language recognition system. In *Proc. Int'l Conf. Pattern Recogn.*, volume 4, pages 849–853, 2000.

[69] F. Jelinek. *Statistical Methods For Speech Recognition.* MIT Press, 1998.

[70] M. Jordan. *An Introduction to Probabilistic Graphical Models.* Draft version, to be published.

[71] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[72] P. Jourlin. Word dependent acoustic-labial weights in HMM-based speech recognition. In *Proc. European Tutorial Workshop on Audio-Visual Speech Processing (AVSP)*, pages 69–72, 1997.

[73] D. Jurafsky and J. Martin. *Speech and language processing: An Introduction to Natural Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall, 2000.

[74] M. Kadous. Machine recognition of Auslan signs using powergloves: Towards large-lexicon recognition of sign language. In *Proc. Workshop the Integration of Gest. in Lang. & Speech*, pages 165–174, 1996.

[75] M. Kadous. Learning comprehensible descriptions of multivariate time series. In *Proc. Int'l Conf. Machine Learning*, pages 454–463, 1999.

[76] A. Kendon. Current issues in the study of gesture. In J.-L. Nespoulous, P. Peron, and A. Lecours, editors, *The Biological Foundation of Gestures: Motor and Semiotic Aspects*, pages 23–47, Hillsdale, NJ, 1986. Lawrence Erlbaum Associates.

[77] A. Kendon. How gestures can become like words. In F. Poyatos, editor, *Cross-Cultural Perspectives in Nonverbal Communication*, pages 131–141, Toronto, 1988. Hogrefe.

[78] A. Kendon. Human gesture. In T. I. K.R. Gibson, editor, *Tools, language, and cognition in human evolution*, pages 43–62. Cambridge University Press, 1993.

[79] R. Kennaway. Experience with and requirements for a gesture description language for synthetic animation. In *Proc. Gesture Workshop*, pages 300–311, 2003.

[80] J.-S. Kim, W. Jang, and Z. Bien. A dynamic gesture recognition system for the Korean Sign Language (KSL). *IEEE Trans. Syst., Man Cybern., Part B: Cybern.*, 26(2):354–359, Apr 1996.

[81] E. Klima and U. Bellugi. *The Signs of Language.* Harvard Univ. Press, Cambridge, Mass., 1979.

[82] T. Kobayashi and S. Haruyama. Partly-hidden Markov model and its application to gesture recognition. In *Proc Int'l Conf. Acoustics, Speech and Signal Processing*, volume 4, pages 3081–3084, 1997.

[83] A. Koizumi, H. Sagawa, and M. Takeichi. An annotated Japanese Sign Language corpus. In *Proc. Int'l Conf. Language Resources and Evaluation*, volume III, pages 927–930, 2002.

[84] D. Koller and N. Friedman. Bayesian Networks and Beyond. Draft version, to be published.

[85] W. W. Kong and S. Ranganath. 3-D hand trajectory recognition for Signing Exact English. In *Proc. Int'l Conf. Auto. Face & Gest. Recogn.*, pages 535–540, 2004.

[86] G. A. Korn and T. M. Korn. *Mathematical Handbook for Scientists and Engineers*. McGraw-Hill, New York, 2nd edition, 1968.

[87] J. Kramer and L. Leifer. The talking glove: An expressive and receptive verbal communication aid for the deaf, deaf-blind, and nonvocal. In *Proc. 3rd Annu. Conf. Computer Tech., Special Education, Rehab.*, pages 335–340, 1987.

[88] S. Kumar and M. Hebert. Discriminative fields for modeling spatial dependencies in natural images. In S. Thrun, L. Saul, and B. Scholkopf, editors, *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2003. MIT Press.

[89] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*, pages 282–289, 2001.

[90] S. Lühr, H. H. Bui, S. Venkatesh, and G. A. W. West. Recognition of human activity through hierarchical stochastic learning. In *Proc. First IEEE Int'l Conf. Pervasive Computing and Comm.*, 2003.

[91] C.-H. Lee and J.-L. Gauvain. Speaker adaptation based on MAP estimation of HMM parameters. *IEEE Int'l Conf. Acoustics, Speech and Signal Processing*, 2:558–561, Apr 1993.

[92] X. Lei, G. Ji, T. Ng, J. Bilmes, and M. Ostendorf. DBN-based multi-stream models for Mandarin toneme recognition. In *IEEE Intl Conf Acoustics, Speech and Signal Processing*, Mar 2005.

[93] R.-H. Liang and M. Ouhyoung. A real-time continuous gesture recognition system for sign language. In *Proc. Int'l Conf. Auto. Face & Gest. Recog*, pages 558–565, 1998.

[94] S. Liddell. *Grammar, Gesture, and Meaning in American* Sign Language. Cambridge Univ. Press, Cambridge, 2003.

[95] S. Liddell and R. Johnson. American Sign Language: The phonological base. *Sign Language Studies*, 64:195–277, 1989.

[96] Y. Liu, J. Carbonell, P. Weigele, and V. Gopalakrishnan. Segmentation conditional random fields (SCRFs): A new approach for protein fold recognition. In *ACM Intl. Conf. on Research in Computational Molecular Biology (RECOMB05)*, 2005.

[97] S. Lu, S. Igi, H. Matsuo, and Y. Nagashima. Towards a dialogue system based on recognition and synthesis of Japanese Sign Language. In *Proc. Gesture Workshop*, pages 259–271, 1997.

[98] J. Luettin, G. Potamianos, and C. Neti. Asynchronous stream modeling for large vocabulary audio-visual speech recognition. In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, page 169172, 2001.

[99] Y. Luo and J.-N. Hwang. Video sequence modeling by dynamic Bayesian networks: a systematic approach from coarse-to-fine grains. In *Proc. 2003 Int'l Conf. Image Processing*, volume 2, pages 615–618, Sep 2003.

[100] H. Matsuo, S. Igi, S. Lu, Y. Nagashima, Y. Takata, and T. Teshima. The recognition algorithm with non-contact for Japanese Sign Language using morphological analysis. In *Proc. Gesture Workshop*, pages 273–285, 1997.

[101] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Seventh Conf. on Natural Language Learning (CoNLL)*, 2003.

[102] I. McCowan, D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard, and D. Zhang. Automatic analysis of multimodal group actions in meetings. *IEEE Trans. on Patt. Analysis and Mach. Intell.*, 27:305317, 2005.

[103] R. McGuire, J. Hernandez-Rebollar, T. Starner, V. Henderson, H. Brashear, and D. Ross. Towards a one-way American Sign Language translator. In *Proc. Int'l Conf. Auto. Face & Gest. Recog.*, pages 620–625, 2004.

[104] D. McNeill. *Hand and Mind: What Gestures Reveal about Thought.* Univ. of Chicago Press, Chicago, 1992.

[105] C. Miyajima, K. Tokuda, and T. Kitamura. Audio-visual speech recognition using MCE-based HMMs and model-dependent stream weights. In *Proc. Int'l Conf. on Spoken Language Processing*, volume II, page 10231026, 2000.

[106] A. Morris, A. Hagen, H. Glotin, and H. Bourlard. Multi-stream adaptive evidence combination for noise robust asr. *Speech Communication*, 2001.

[107] K. Murakami and H. Taguchi. Gesture recognition using recurrent neural networks. In *Proc. SIGCHI Conf. Human Factors in Computing Syst.*, pages 237–242, 1991.

[108] K. Murphy. *Bayes Net Toolbox for Matlab.* http://bnt.sourceforge.net/.

[109] K. Murphy. Dynamic Bayesian networks. In *[70]*.

[110] K. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning.* PhD thesis, UC Berkeley, Computer Science Division, 2002.

[111] K. Murphy. *Hierarchical HMMs.* Technical Report (http://www.cs.ubc.ca/ murphyk/papers.html#techreports), Nov 2002.

[112] K. Murphy and Y. Weiss. The factored frontier algorithm for approximate inference in DBNs. In *UAI*, 2001.

[113] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: an empirical study. In *UAI*, 1999.

[114] S. Nakamura, H. Ito, and K. Shikano. Stream weight optimization of speech and lip image sequence for audiovisual speech recognition. In *Proc. Int'l Conf. on Spoken Language Processing*, volume III, page 2023, 2000.

[115] A. Nefian, L. Liang, X. Pi, X. Liu, and K. Murphy. Dynamic Bayesian networks for audio-visual speech recognition. *EURASIP Journal on Applied Signal Processing*, 11:1274–1288, 2002.

[116] C. Neidle, S. Sclaroff, and V. Athitsos. Signstream: a tool for linguistic and computer vision research on visual-gestural language data. *Behaviour Research Methods, Instruments & Computers*, 33(3):311–32, 2001.

[117] C. Neti, G. Potamianos, J. Luettin, I. Matthews, H. Glotin, D. Vergyri, J. Sison, A. Mashari, and J. Zhou. *Audio visual speech recognition, Final workshop 2000 report.* Tech. Rep., Center for Language and Speech Processing, The Johns Hopkins University, Baltimore, Md, USA, 2000.

[118] N. T. Nguyen, D. Q. Phung, S. Venkatesh, and H. Bui. Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model. In *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, CVPR 2005*, volume 2, pages 955–960, June 2005.

[119] H. Nock and M. Ostendorf. Parameter reduction schemes for loosely coupled HMMs. *Compute Speech and Language*, 17:233–262, 2003.

[120] S. Okawa, T. Nakajima, and K. Shirai. A recombination strategy for multiband speech recognition based on mutual information criterion. In *Proc. European Conf. on Speech Comm. and Tech. (EUROSPEECH)*, volume 2, pages 603–606, 1999.

[121] N. Oliver, E. Horvitz, and A. Garg. Layered representations for human activity recognition. In *Proc. of the Fourth IEEE Int'l Conf. on Multimodal Interfaces, (ICMI'02)*, 2002.

[122] N. Oliver, B. Rosario, and A. Pentland. A Bayesian computer vision system for modeling human interaction. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(8):831–843, Aug 2000.

[123] S. Park and J. Aggarwal. A hierarchical Bayesian network for event recognition of human actions and interactions. *Multimedia Systems*, 10:164–179, 2004.

[124] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[125] D. Perlmutter. Sonority and syllable structure in american sign language. In G. Coulter, editor, *Phonetics and Phonology: Current issues in ASL phonology*, volume 3, pages 227–261, San Diego, 1993. Academic Press.

[126] H. Poizner, E. Klima, U. Bellugi, and R. Livingston. Motion analysis of grammatical processes in a visual-gestural language. In *Proc. ACM SIGGRAPH/SIGART Interdisciplinary Workshop*, pages 271–292, 1983.

[127] G. Potamianos, C. Neti, J. Luettin, and I. Matthews. Audio-visual automatic speech recognition: An overview. In G. Bailly, E. Vatikiotis-Bateson, and P. Perrier, editors, *Issues in Visual and Audio-Visual Speech Processing.* MIT Press, 2004.

[128] F. Quek. Toward a vision-based hand gesture interface. In *Proc. Virtual Reality Sofware & Tech. Conf.*, pages 17–29, 1994.

[129] F. Quek. Eyes in the interface. *Image and Vision Computing*, 13(6), Aug 1995.

[130] F. Quek. Hidden Conditional Random Fields. *IEEE Trans. Pattern Anal. Machine Intell.*, 29(10):1848 – 1852, Oct 2007.

[131] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Procs. of the IEEE*, 77(2), Feb 1999.

[132] D. Rubin. Using the SIR algorithm to simulate posterior distributions. In J. Bernardo, M. DeGroot, D. Lindley, and A. Smith, editors, *Bayesian Statistics 3*, pages 395–402. Oxford University Press, 1988.

[133] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2nd edition, 2002.

[134] H. Sagawa and M. Takeuchi. A method for analyzing spatial relationships between words in sign language recognition. In *Proc. Gesture Workshop*, pages 197 – 210, 1999.

[135] H. Sagawa and M. Takeuchi. Development of an information kiosk with a sign language recognition system. In *Proc. ACM Conf. Universal Usability*, pages 149–150, 2000.

[136] H. Sagawa and M. Takeuchi. A method for recognizing a sequence of sign language words represented in a Japanese sign language sentence. In *Proc. Int'l Conf. Auto. Face & Gest. Recog.*, pages 434–439, 2000.

[137] W. Sandler. *Phonological representation of the sign*. Foris, Dordrecht, 1989.

[138] K. Sato and Y. Sakakibara. Rna secondary structural alignment with conditional random fields. *Bioinformatics*, 21(ii):237242, 2005.

[139] B. Settles. Abner: an open source tool for automatically tagging genes, proteins, and other entity names in text. *Bioinformatics*, 21(14):31913192, 2005.

[140] F. Sha and F. Pereira. Shallow parsing with conditional random fields. In *Proc. HLT-NAACL*, page 213220, 2003.

[141] M. Skounakis, M. Craven, and S. Ray. Hierarchical hidden Markov models for information extraction. In *IEEE/RSJ Int't Conf. Intelligent Robots and Systems (IROS)*, 2003.

[142] T. Starner. *Visual Recognition of American Sign Language Using Hidden Markov Models*. MIT S.M. Thesis, Feb 1995.

[143] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Trans. Pattern Anal. Machine Intell.*, 20(12):1371–1375, Dec 1998.

[144] W. Stokoe. Sign language structure: an outline of the visual communication system of the American deaf. In *Studies in Linguistics: Occasional Papers 8*, Silver Spring, MD, 1960. Revised 1978. Lindstok Press.

[145] M.-C. Su. A fuzzy rule-based approach to spatio-temporal hand gesture recognition. *IEEE Trans. Syst., Man Cybern., Part C: Applicat. Rev.*, 30(2):276–281, May 2000.

[146] M.-C. Su, Y.-X. Zhao, H. Huang, and H.-F. Chen. A fuzzy rule-based approach to recognizing 3-d arm movements. *IEEE Trans. Neural Syst. Rehab. Eng.*, 9(2):191–201, June 2001.

[147] A. Sutherland. Real-time video-based recognition of sign language gestures using guided template matching. In *Proc. Gesture Workshop*, pages 31–38, 1996.

[148] S. Tamura and S. Kawasaki. Recognition of sign language motion images. *Pattern Recogn.*, 21(4):343–353, 1988.

[149] N. Tanibata, N. Shimada, and Y. Shirai. Extraction of hand features for recognition of sign language words. In *Proc. Int'l Conf. Vision Interface*, pages 391–398, 2002.

[150] J.-C. Terrillon, A. Pilpr, Y. Niwa, and K. Yamamoto. Robust face detection and Japanese sign language hand posture recognition for human-computer interaction in an “Intelligent” room. In *Proc. Int'l Conf. Vision Interface*, pages 369–376, 2002.

[151] M. Tomlinson, M. Russell, and N. Brooke. Integrating audio and visual information to provide highly robust speech recognition. In *Proc. IEEE Int'l Conf. Acoustics, Speech, Signal Processing*, page 821824, 1996.

[152] C. Valli and C. Lucas. *Linguistics of American sign language: a resource text for ASL users.* Gallaudet Univ. Press, Washington, D.C., 1992.

[153] P. Vamplew. *Recognition of Sign Language Using Neural Networks.* PhD Thesis, Dept. of Computer Science, Univ. of Tasmania, Washington, D.C., May 1996.

[154] P. Vamplew and A. Adams. Recognition of sign language gestures using neural networks. *Australian J. Intell. Info. Processing Syst.*, 5(2):94–102, 1998.

[155] P. Varga and R. Moore. Hidden Markov model decomposition of speech and noise. In *Proc. Int'l Conf. on Acoustics, Speech and Signal Processing*, pages 845–848, 1990.

[156] D. Vergyri. *Integration of Multiple Knowledge Sources in Speech Recognition Using Minimum Error Training.* PhD Thesis, Center for Speech and Language Processing, The Johns Hopkins University, Baltimore, MD, 2000.

[157] C. Vogler. *American Sign Language Recognition: Reducing the Complexity of the Task with Phoneme-based Modeling and Parallel Hidden Markov Models.* PhD thesis, Univ. of Pennsylvania, 2003.

[158] C. Vogler and D. Metaxas. A framework for recognizing the simultaneous aspects of American Sign Language. *Computer Vision Image Understanding*, 81:358–38, 2001.

[159] M. Waldron and S. Kim. Isolated ASL sign recognition system for deaf persons. *IEEE Trans. Rehabilitation Eng.*, 3(3):261–271, Sep 1995.

[160] C. Wang, W. Gao, and J. Ma. An approach to automatically extracting the basic units in Chinese Sign Language. In *Proc. Int'l Conf Signal Processing, ICSP2000*, volume 2, pages 855–858, 2000.

[161] C. Wang, W. Gao, and S. Shan. An approach based on phonemes to large vocabulary Chinese Sign Language recognition. In *Proc. Int'l Conf. Auto. Face & Gest. Recog.*, pages 393–398, 2002.

[162] T. Watanabe and M. Yachida. Real time gesture recognition using eigenspace from multi input image sequence. In *Proc. Int'l Conf. Auto. Face & Gest. Recog.*, pages 428–43, 1998.

[163] A. Wexelblat. Research challenges in gesture: Open issues and unsolved problems. In *Proc. Gesture Workshop*, pages 1–12, 1997.

[164] R. Wilbur. Syllables and segments: Hold the movement and move the holds! In G. Coulter, editor, *Phonetics and Phonology: Current issues in ASL phonology*, volume 3, pages 135–168, San Diego, 1993. Academic Press.

[165] A. Wilson and A. Bobick. Parametric hidden Markov models for gesture recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, 21(9):885–900, 1999.

[166] P. Woodland. Speaker adaptation: Techniques and challenges. *Automatic Speech Recognition and Understanding Workshop*, 1:85–90, 1999.

[167] J. Wu and W. Gao. A fast sign word recognition method for Chinese Sign Language. In *Proc. Int'l Conf. Advances in Multimodal Interfaces*, pages 599–606, 2000.

[168] J. Wu and W. Gao. The recognition of finger-spelling for Chinese Sign Language. In *Proc. Gesture Workshop*, pages 96–100, 2001.

[169] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun. Unsupervised discovery of multilevel statistical video structures using hierarchical hidden Markov models. In *Proc. IEEE ICME*, 2003.

[170] Q. J. Y. Zhang. Active and dynamic information fusion for facial expression understanding from image sequences. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(5):699–714, May 2005.

[171] M.-H. Yang, N. Ahuja, and M. Tabb. Extraction of 2d motion trajectories and its application to hand gesture recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(8):1061–1074, Aug 2002.

[172] T. Yoshikawa. *Foundations of robotics: analysis and control.* The MIT Press, Cambridge, Massachusetts, 1990.

[173] S. Young. A review of large-vocabulary continuous-speech recognition. *IEEE Sig. Processing Mag*, pages 45–57, Sep 1996.

[174] Q. Yuan, W. Gao, H. Yao, and C. Wang. Recognition of strong and weak connection models in continuous sign language. In *Proc. Int'l Conf. Pattern Recogn.*, volume 1, pages 75–78, 2002.

[175] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Modeling individual and group actions in meetings with layered HMMs. *IEEE Trans. Multimedia*, 8(3):509–520, June 2006.

[176] Y. Zhang, Q. Diao, S. Huang, W. Hu, C. Bartels, and J. Bilmes. DBN based multi-stream models for speech. In *IEEE Intl Conf Acoustics, Speech and Signal Processing*, Apr 2003.

[177] Y. Zhang, S. Levinson, and T. Huang. Speaker independent audio-visual speech recognition. In *IEEE Int'l Conf. on Multimedia and Expo*, volume 2, page 10731076, 2000.

[178] T. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill. Hand gesture interface device. In *Proc SIGCHI/GI Conf. Human factors in Computing Syst. and Graphics Interface*, pages 189–192, 1986.

[179] G. Zweig. *Speech Recognition with Dynamic Bayesian Networks*. PhD thesis, U.C. Berkeley, Dept. Comp. Sci., 1998.

Appendix A

Notation and Terms

This is a list of notations used in the thesis in general. Any specific notations used in specific sections or chapters are defined when they first appear.

- $P(X = x)$: probability of the random variable X taking on the value x. This is generally abbreviated as $P(x)$.
- $|X|$: the number of possible values for X.
- $\mathbf{Pa}_X$: parents of variable X.

Appendix B

List of lexical words and inflections for continuous signing experiments

Table B.1 lists the 29 different lexical words present in the vocabulary. Only a subset of these are combined with an inflection value to form signs, i.e. some signs are formed from a lexical word only, with no inflectional meaning added. Table B.2 lists the 3 different temporal aspect inflection values and Table B.3 lists the 11 different directional verb inflection values used in forming signs. The notation for directional verb inflections show the subject and object that the root verb (notated as a generic 'VERB') identifies through its movement path direction. The terms to the left and right of the arrow are the subject and object, respectively. In the set of sentences containing directional verbs, the subjects and objects that the verb may indicate includes the signer (denoted as 'I'), the addressee (denoted as 'YOU') and two other non-present referents 'GIRL' and 'JOHN'. In sentences that refer to 'GIRL', this referent was established at roughly to the right of the signer, using the

Table B.1: Lexical root words used in constructing signs for the experiments.

Category	Lexical root words
Nouns	BOOK, CAT, EMAIL, GIRL, HOME, JOHN, PAPER, PEN, PICTURE, SIGN_LANGUAGE, TEACH
Pronouns	I, MY, YOU, YOUR, INDEX$^{\rightarrow\mathrm{GIRL}}$, INDEX$^{\rightarrow\mathrm{JOHN}}$
Verbs	BLAME, EAT, GIVE, GO, HELP, LOOK, PRINT, SEND, TAKE, WRONG
Adjectives	A_LOT, BLACK
Other	REST_START, REST_END

Note: INDEX$^{\rightarrow\mathrm{x}}$ is produced with the index finger extended, directed towards the person being referred to, x. For example, INDEX$^{\rightarrow\mathrm{GIRL}}$ points towards GIRL.

Table B.2: Temporal aspect inflections used in constructing signs for the experiments.

[DURATIONAL], [HABITUAL], [CONTINUATIVE]

method mentioned in Section 1.1.2. Similarly, 'JOHN' was established at roughly to the left of the signer.

Table B.4 lists the signs that were left out of the training sentences in the experiments reported in Section 6.8. This set of signs are referred to as unseen signs. The purpose of the experiments were to test the combined model (MH-HMM), with training done on a reduced sign vocabulary.

Table B.3: Directional verb inflections used in constructing signs for the experiments.

$\text{VERB}^{\text{I}\rightarrow\text{YOU}}$, $\text{VERB}^{\text{YOU}\rightarrow\text{I}}$, $\text{VERB}^{\text{I}\rightarrow\text{GIRL}}$, $\text{VERB}^{\text{GIRL}\rightarrow\text{I}}$, $\text{VERB}^{\text{I}\rightarrow\text{JOHN}}$, $\text{VERB}^{\text{JOHN}\rightarrow\text{I}}$, $\text{VERB}^{\text{YOU}\rightarrow\text{GIRL}}$, $\text{VERB}^{\text{GIRL}\rightarrow\text{YOU}}$, $\text{VERB}^{\text{YOU}\rightarrow\text{JOHN}}$, $\text{VERB}^{\text{JOHN}\rightarrow\text{YOU}}$, $\text{VERB}^{\text{GIRL}\rightarrow\text{JOHN}}$

Table B.4: Signs not present in the training sentences in the experiments on training with reduced vocabulary (see Section 6.8).

$\text{HELP}^{\text{I}\rightarrow\text{GIRL}}$, $\text{HELP}^{\text{GIRL}\rightarrow\text{I}}$, $\text{HELP}^{\text{I}\rightarrow\text{JOHN}}$, $\text{HELP}^{\text{JOHN}\rightarrow\text{I}}$, $\text{GIVE}^{\text{I}\rightarrow\text{YOU}}$, $\text{GIVE}^{\text{I}\rightarrow\text{JOHN}}$, $\text{EAT}^{\text{[DURATIONAL]}}$, $\text{EAT}^{\text{[HABITUAL]}}$, $\text{EAT}^{\text{[CONTINUATIVE]}}$, $(\text{GIVE}^{\text{[DURATIONAL]}})^{\text{I}\rightarrow\text{YOU}}$, $(\text{GIVE}^{\text{[HABITUAL]}})^{\text{I}\rightarrow\text{YOU}}$, $(\text{GIVE}^{\text{[CONTINUATIVE]}})^{\text{I}\rightarrow\text{YOU}}$, $(\text{GIVE}^{\text{[DURATIONAL]}})^{\text{I}\rightarrow\text{GIRL}}$, $(\text{GIVE}^{\text{[HABITUAL]}})^{\text{I}\rightarrow\text{GIRL}}$, $(\text{GIVE}^{\text{[DURATIONAL]}})^{\text{I}\rightarrow\text{JOHN}}$, $(\text{GIVE}^{\text{[HABITUAL]}})^{\text{I}\rightarrow\text{JOHN}}$

Appendix C

Position and orientation measurements in continuous signing experiments

Suppose we have an object H, and an orthogonal coordinate frame associated with it. This coordinate frame's origin, and x, y, z axes can be expressed relative to a base coordinate frame as ${}^{B}\underline{\mathrm{o}}_{H}$, ${}^{B}\underline{\mathrm{x}}_{H}$, ${}^{B}\underline{\mathrm{y}}_{H}$, and ${}^{B}\underline{\mathrm{z}}_{H}$, respectively. ${}^{B}\underline{\mathrm{x}}_{H}$, ${}^{B}\underline{\mathrm{y}}_{H}$, and ${}^{B}\underline{\mathrm{z}}_{H}$ are the columns of a rotation matrix ${}^{B}\mathrm{R}_{H}$, that maps a vector expressed relative to object H's coordinate frame to another vector expressed relative to the base coordinate frame. For example, the x-axis in H's coordinate frame is $[1\ 0\ 0]^{T}$. To express it relative to the base coordinate system, we apply the rotation matrix ${}^{B}\mathrm{R}_{H}$ to get,

$$
\begin{aligned}
{}^{B}\mathrm{R}_{H}\begin{bmatrix}1\\0\\0\end{bmatrix} &= \begin{bmatrix}{}^{B}\underline{\mathrm{x}}_{H} & {}^{B}\underline{\mathrm{y}}_{H} & {}^{B}\underline{\mathrm{z}}_{H}\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix} \\
&= {}^{B}\underline{\mathrm{x}}_{H} \qquad \text{(C.1)}
\end{aligned}
$$

The case for ${}^{B}\underline{\mathrm{y}}_{H}$ and ${}^{B}\underline{\mathrm{z}}_{H}$ is analogous.

If we have another object W, with an attached coordinate frame whose origin and axes are expressed relative to the base coordinate frame as ${}^{B}\underline{\mathrm{o}}_{W}$, ${}^{B}\underline{\mathrm{x}}_{W}$, ${}^{B}\underline{\mathrm{y}}_{W}$, and ${}^{B}\underline{\mathrm{z}}_{W}$, we need to express the origin and axes of H's coordinate frame relative to W's coordinate frame, i.e. ${}^{W}\underline{\mathrm{o}}_{H}$, ${}^{W}\underline{\mathrm{x}}_{H}$, ${}^{W}\underline{\mathrm{y}}_{H}$, and ${}^{W}\underline{\mathrm{z}}_{H}$.

${}^{W}\underline{\mathrm{x}}_{H}$, ${}^{W}\underline{\mathrm{y}}_{H}$, and ${}^{W}\underline{\mathrm{z}}_{H}$ are the columns of the rotation matrix ${}^{W}\mathrm{R}_{H}$, that maps a vector expressed relative to object H's coordinate frame to one expressed relative to object W's coordinate frame. This rotation matrix can be computed as two rotations applied successively [172], i.e.

$$
\begin{aligned}
{}^{W}\mathrm{R}_{H} &= {}^{W}\mathrm{R}_{B}.{}^{B}\mathrm{R}_{H} \\
&= \left({}^{B}\mathrm{R}_{W}\right)^{T}.{}^{B}\mathrm{R}_{H} \qquad \text{(C.2)}
\end{aligned}
$$

which is straightforward to compute since we know both ${}^{B}\mathrm{R}_{W}$ and ${}^{B}\mathrm{R}_{H}$.

Next, we find ${}^{W}\underline{\mathrm{o}}_{H}$ by first finding the difference vector between the origins of the H and W coordinate frames, ${}^{B}\underline{p} =^{B} \underline{\mathrm{o}}_{H} -^{B} \underline{\mathrm{o}}_{W}$. As this difference vector is expressed relative to the base coordinate frame, so we need to apply a rotation matrix ${}^{W}\mathrm{R}_{B}$, in order to express it relative to W's coordinate frame. Thus,

$$\begin{aligned} {}^{W}\underline{\mathrm{o}}_{H} &= {}^{W}\mathrm{R}_{B}.{}^{B}\underline{p} \\ &= \left({}^{B}\mathrm{R}_{W}\right)^{T}.\left({}^{B}\underline{\mathrm{o}}_{H} -{}^{B} \underline{\mathrm{o}}_{W}\right) \end{aligned} \tag{C.3}$$

which is again straightforward to compute since we know all the terms in the last expression.

In the experiments of Chapter 6, data capture was through the Polhemus tracker which reported 3-dimensional position and orientation of the right hand and waist sensors, relative to the transmitter frame. Conceptually, each of the sensors has an attached orthogonal coordinate frame. The reported 3-dimensional position data is the x, y, and z coordinates of its origin, relative to the transmitter frame. If we denote the transmitter frame as the base frame (B), and right hand and waist sensors as H and W, respectively, their reported 3-dimensional positions correspond to the terms ${}^{B}\underline{\mathrm{o}}_{H}$ and ${}^{B}\underline{\mathrm{o}}_{W}$ mentioned in the above paragraphs. The reported orientation angles are the roll (${}^{B}\psi_{i}$), pitch (${}^{B}\theta_{i}$) and yaw (${}^{B}\phi_{i}$) (for $i = H, W$) of the sensor's coordinate frame, relative to the transmitter frame. The yaw, pitch and row angles are equivalent to the angles of successive rotations about the z, y and x axes of the reference frame [86]. Thus we can calculate the corresponding rotation matrix ${}^{W}\mathrm{R}_{i}$ as,

$$
{}^{B}\mathrm{R}_i = \begin{bmatrix} cos({}^{B}\psi_i) & -sin({}^{B}\psi_i) & 0 \\ sin({}^{B}\psi_i) & cos({}^{B}\psi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} cos({}^{B}\theta_i) & 0 & sin({}^{B}\theta_i) \\ 0 & 1 & 0 \\ -sin({}^{B}\theta_i) & 0 & cos({}^{B}\theta_i) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos({}^{B}\phi_i) & -sin({}^{B}\phi_i) \\ 0 & sin({}^{B}\phi_i) & cos({}^{B}\phi_i) \end{bmatrix} \quad \text{for } i = H, W \tag{C.4}
$$

For $i = H$, the rotation matrix obtained from equation (C.4) is ${}^{B}\mathrm{R}_H$, for $i = W$, the rotation matrix obtained is ${}^{B}\mathrm{R}_W$. We can now apply equation (C.2) to obtain ${}^{W}\mathrm{R}_H$ and thus the orientation of the right hand sensor relative to the waist sensor's coordinate frame in terms of the three axes, ${}^{W}\underline{\mathrm{x}}_H$, ${}^{W}\underline{\mathrm{y}}_H$, and ${}^{W}\underline{\mathrm{z}}_H$. We can also apply equation (C.3) to obtain ${}^{W}\underline{\mathrm{o}}_H$, the position of the right hand sensor relative to the waist sensor's coordinate frame . Note that in Chapter 6, the superscript W is dropped when referring to the right hand sensor's coordinate frame.

BEYOND LEXICAL MEANING: PROBABILISTIC MODELS FOR SIGN LANGUAGE RECOGNITION

SYLVIE C.W. ONG

NATIONAL UNIVERSITY OF SINGAPORE

2007

Beyond Lexical Meaning: Probabilistic Models **Sylvie C.W. Ong** **2007**

for Sign Language Recognition