NAIST-IS-DT0161006

Doctor's Thesis

Statistical Learning from Multiple Information Sources

Masashi Inoue

March 24, 2004

Department of Information Systems
Graduate School of Information Science
Nara Institute of Science and Technology

Doctor's Thesis
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
DOCTOR of SCIENCE

Masashi Inoue

Thesis Committee: Shin Ishii, Professor
Noboru Sugamura, Professor
Yuji Matsumoto, Professor
Naonori Ueda, Associate Professor

Statistical Learning from Multiple Information Sources*

Masashi Inoue

Abstract

In intelligent information processing tasks such as pattern recognition and information retrieval (IR), probabilistic models are now widely used because they can represent ambiguities of observed data and are robust against noises. Parameters of probabilistic models are statistically estimated (learned) from given training data. However, when the training data contain an insufficient amount of information, the learned model becomes unreliable and its performance severely deteriorates. This thesis proposes two novel learning algorithms that use multiple information sources to mitigate this information scarcity problem in the following two applications.

The first application is solving classification problems in which optimal class labels are automatically assigned to observations whose class labels are unknown. Among various types of classification problems, this paper considers classification of sequences that consist of sequential observation points. As a classifier, we focus on the hidden Markov model (HMM), which has been widely used for the classification of sequences. Generally, an HMM is trained on labeled data that consist of observed feature values and class labels. However, due to the high labeling cost, the amount of labeled training data is often small. In this thesis, we propose a learning scheme called semi-supervised learning to improve the classification performance even if the amount of labeled training data is small. The proposed scheme uses both of a small amount of labeled data, and unlabeled data that are not usually used for learning. First, we design a suitable HMM structure for using the unlabeled sequences. Then, we formally derive a semi-supervised learning algorithm in which the convergence property is theoretically guaranteed. Next, we apply the proposed method to two types of time series sequences: those acquired from sign language sign data and those acquired from speech phoneme data. Experimental results show that the proposed method outperforms conventional methods.

The second application is solving IR problems, especially cross-media IR in which queries and corresponding target data belong to different media. More specifically, this thesis focuses on situations where queries are represented as text and target data are images. When textual annotations explaining the contents of images are provided, such cross-media image retrieval can be regarded as ordinary textual IR. In text retrieval, associations between words can be learned from data and used to relate queries and text in the documents. However, in image retrieval, the number of annotations is too small to learn the word relationships. Using the fact that annotations and images are related, we regard images as the paired data of annotations.

*Doctor's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DT0161006, March 24, 2004.

We propose a method for estimating similarities among annotation words by using the paired image data to interpolate sparseness of annotation data. We apply the proposed method to the retrieval of photo images and show the usefulness of the paired data for improving retrieval performance. We also compare the proposed method with conventional methods to show its advantages.

Keywords:

statistical learning, multiple information sources, classification of sequences, cross-media information retrieval

複数情報源からの統計的学習*

井上 雅史

内容梗概

パターン認識，情報検索などの知的情報処理において，曖昧さの表現に適することやノイズに対する頑健さ等の理由から，現在，確率モデルアプローチが広く用いられている．確率モデルのパラメータはデータから統計的に推定（学習）される．その際，学習に必要な訓練データが十分な情報を含んでいない場合，学習後のモデルの精度は不十分となり，利用される情報処理課題において十分な性能を発揮することができない．この問題に対し，本論文では，以下の二つの応用領域において，複数の情報源を利用することで訓練データの情報不足を補う手法について検討する．

第一の応用領域として，未知のデータが既知のどのクラスに所属するかを判定する識別問題を考える．識別問題の中でも，複数の観測点からなる系列データを対象とし，確率モデルとして，系列データの識別に広く用いられる隠れマルコフモデル（HMM）に焦点をあてる．HMM の学習においては，観測された特徴量とクラスラベルの組であるラベル有りデータが通常訓練データとして用いられるが，ラベル付けにコストがかかるため，しばしば訓練データが不足する．そこで本論文では，これまで利用されてこなかったラベル無しデータを，少数のラベル有りデータと併用する半教師有り学習により，未学習データに対する識別性能を向上させる手法を提案する．ラベル無しデータの利用に適した HMM のモデル構造を考案し，収束性を理論保証する半教師有り学習アルゴリズムを導出する．提案手法を手話単語の識別問題および音素の識別問題に適用し，既存法と提案手法とを実験的に比較し，提案手法の優位性を確認した．

第二の応用領域として，情報検索問題，特に，問い合わせが検索対象と異なるメディアである，クロスメディア検索問題を対象とする．具体的には，問い合わせが単語で，検索対象が画像であるクロスメディア画像検索問題を取り扱う．画像に対して単語による内容説明（アノテーション）が付与されている場合，クロスメディア画像検索を一般的なテキスト検索と同種の枠組みで捉えることができる．テキスト検索では，問い合わせとアノテーションを関連づけるために，データから推定した単語間の類似関係を利用する．しかし，アノテーションは通常のテキストと比較して量が非常に少なく，単語間関係を学習するには不十分である．本研究ではアノテーションと画像が対になっていることに着目し，画像をアノテーションの関連データとして取り扱い，アノテーションからのみでは学習が困難な単語間関係を画像情報を用いて補間する手法を提案する．提案手法を物体写真画像の検索に適用し，関連データの利用が検索精度を向上させることを確認した．また，既存手法に対する優位性についても考察した．

*奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 博士論文, NAIST-IS-DT0161006, 2004 年 3 月 24 日.

キーワード

統計的学習 複数情報源 系列データ識別 クロスメディア情報検索

Acknowledgements

I am greatful to my advisor Naonori Ueda for his guidances and discussions leading me to many important concepts in my research. Learning from his sense of realism and uncompromising attitude toward research has been the most rewarding part of my experience doing graduate work. This thesis could not have been completed without his enormous investments of time and effort in me. I would also like to thank Shin Ishii for his support in all aspects of my graduate life. Deep appreciation goes to Yuji Matsumoto for being my committee member and giving his insightful comments on my work. The Laboratory for Communication Studies is operated by the NTT Communication Science Laboratories under a cooperative agreement with NAIST. I would like to thank the laboratories for providing me with the opportunity to do my graduate studies there and to Noboru Sugamura for his cooperation. Thanks also goes to Yoh'ichi Tohkura and Ryohei Nakano, former leaders of the laboratory.

My research could not have been done without using real world data. Sincere thanks go to the following people who helped me in this most essential part of the research: Hitoshi Tomita for the construction of and his advice on gesture data used in Chapter 2, Erik McDermott for his help on speech data also used in in Chapter 2, and Mark Steyvers for his help on image data used in Chapter 3.

The encouragement and support from previous and present members of the Laboratory for Communication Studies and the Theoretical Life Science Laboratory are highly appreciated. In particular, I would like to thank Hiroyuki Kubotani for his friendship while I was a master course student. Shigeyuki Oba has been my role model from the beginning of my life as a graduate student. Aya Tanimoto helped me with administrative matters in spite of some complications due to my sometimes irregular research activities. Thanks also to all my fellow graduate students from the other research groups at NAIST for providing support toward the completion of this thesis and my graduate degree. Most notably, it has been a great pleasure to work with my knowledgeable colleague Satomi Higuchi on the project work. Faculty members who made me realize how much I had not learned as an undergraduate student are also appreciated. The sincere attitude of the administration staff at NAIST, contrary to what one typically expects of officials, also deserves to be praised. Thanks are also given to my friends in the NTT Communication Science Laboratories, particularly, present and former members of the Emergent Learning Research Group. I am fortunate to have had Yuji Kaneda in the group, since he has the similar research interests to mine. The frequent exchange of information and opinions on various aspects of machine leaning has been fruitful. I found the talk about research activities and others with Sigeru Katagiri and Tomohiro Nakatani provocative and interesting. I would like to recognize Atsushi Nakamura, Shoko Araki, and Yasuhiro Minami for their wisdom along with the implementation tips they provided as needed during the research on speech data. I also wish to credit Taira Hirotoshi and Jun Suzuki for facilitating my research on natural language processing. Additional thanks go to Shinji Watanabe for sharing common basic values with me. Thanks also goes to Hiromichi Suetani for making me sense a crisis in my abilities and attitudes about my academic career during the relaxed chats about many things.

Apart from my laboratories, I am thankful to fellow researchers in my field for the unmatched educational opportunities they gave me and the valuable comments I have received on my presentations and publications. I am indebted to the editors and referees of publications for their generous sharing of insights and knowledge.

I am grateful to the following financial sources for enabling me to pursue my study. NTT Communication Science Laboratories supported my work with a wonderful research environment, equipment and publication expenses. I thank the NAIST Bio-COE program for a research assistantship for the final year of my doctoral study. Financial assistance for our independent brain imaging project has been generously provided by the NAIST Information Science Fund for Young Researchers in the form of a research grant. I would like to acknowledge travel grants from the following organizations: Inoue Foundation for Science, Foundation for C&C Promotion, and Japanese Neural Network Society.

Thanks must also go to my friends outside of academia for helping me to keep my sanity, for showing me what hard work really means, and for being my interface with society. Finally, I extend my deepest gratitude to my family for their patience and understanding throughout my academic endeavors.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Motivation and Definition

This thesis addresses the problem of information insufficiency in learning. Suppose we have observed or accumulated data. Then, *learning* is an inference of the generating mechanism of a set of given data. Such a mechanism is represented by a *model*, and the goal of learning a model is to precisely fit the model to not only the given training data, but also to unseen data as far as possible. If an infinite amount of data were available, the model could be successfully fit. In practice, however, the amount of available training data are limited because data collection is often difficult. Instead, in this thesis, to solve this information insufficiency problem we try to use additional information sources that have different characteristics to the original information but can easily be collected. However, it is not clear how to simultaneously use such additional information sources when leaning the model. The goal of this thesis is to propose practical learning methods to effectively utilize additional information sources.

Broadly speaking, there are two types of models: deterministic models and probabilistic models. In this thesis, we focus on *probabilistic models* because they can represent both the uncertainty of data-generating mechanisms and noises on measurements. To gain a more intuitive understanding, we explain the difference between deterministic and probabilistic models by using Fig. 1.1.

Deterministic models define the rigid relationships between their sub-components or variables. An example is rule-based models consisting of if-then rules [1]. When we observe a list of symbols several times as shown in Fig. 1.1, we can expect that the first and the third symbols will be "a" while the second and the remaining symbols will be "b." The advantage is that these rules are expressive and easy to understand for humans. The disadvantage is that exceptions or noises cannot be dealt with appropriately. The third observation is an example of such irregularities that violates the rule: `if in the third position, then the symbol is ‘‘a.’’` Deterministic models are not suitable for dealing with such ambiguities.

The real world environment usually contains such uncertainties in underlying mechanism

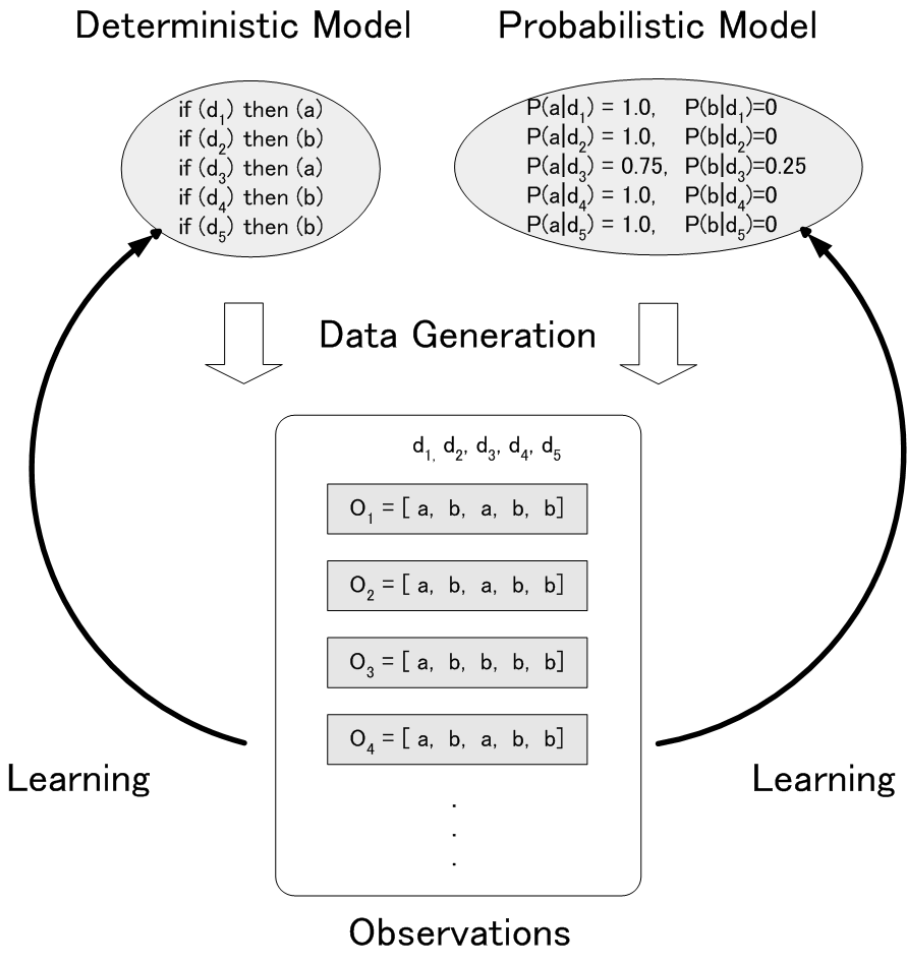


Figure 1.1. When we observe series of data, we can infer unobserved models (either deterministic or probabilistic) that explain how the observations are generated. In this example, we have four observations $o_1, o_2, ..., o_4$, and each observation fills five slots $d_1, d_2, ..., d_5$ by one of two types of symbols ("a" or "b"). An example of a deterministic model is a set of rules, while a probabilistic model may consist of conditional probabilities.

or noises on measurements. These irregularities may be modeled well by probabilistic models. They represent the events of interest by the probabilities of their occurrences; that is, observed data are assumed to be generated from an unknown stochastic process or a probabilistic density function (pdf).

The learning of probabilistic models is called *statistical learning* since it is done with the statistical properties of data. In the example of Fig. 1.1, the probabilities of "a" would be estimated as 1.0 for the first position d_1 and 0.75 for the third position d_3, and the probabilities of "b" would be estimated as being 1.0 for the second position d_2, 0.25 for the third position d_3, and 1.0 for the remaining positions. Uncertainty in the third position can be expressed in this model. Although probabilistic models are powerful representations, here arises a critical question in statistical learning: how reliable are these estimates? For instance, can we believe that the symbols occurring at the fifth position will continue to be "b" forever? In other words, the question on the use of probabilistic models is how we can accurately learn the models.

1.2. Statistical Learning under Information Insufficiency

Roughly speaking, in probabilistic models, the degree of reliability of learned values depends on the amount of information used. In practice, the amount of data available for the learning, called *training data*, is frequently insufficient. This is the fundamental problem in statistical learning and what this thesis addresses. Information insufficiency causes a problem generally termed *over-fitting*, where the learned model that has fitted the insufficient amount of training data perfectly usually fails to represent data we have not yet observed [2]. That is, explaining observed data does not mean explaining unseen data in general. The standard approach to the over-fitting problem follows the *self-help* principle.

Self-help methods use the available information at hand ingeniously so that adverse effects of information insufficiency are reduced. Three major approaches are available: model and data *simplification*, model *ensembles*, and parameter *smoothing*. These methods contrast with the methods that employ multiple information sources, the theme of this thesis. For the sake of comparison, we briefly review them. Simplification can be performed on the models and on data. If a model has a small number of parameters, the amount of data needed for estimation will also be small. Various information criteria have been proposed to select model size. If the dimensionality of data becomes small, the number of parameters also becomes small. Feature extraction and feature selection techniques are used in this situation. The former method includes the use of principal component analysis (PCA) [3], [4] and latent semantic indexing (LSI) [5] for example. A comparative study was conducted on various feature (word) selection techniques in text classification [6]. Another example for comparative study concerns the ranking of features according to their saliency [7]. Ensemble techniques combine different models learned from different initial parameter values or subsets of the data. Consequently, the estimation of parameters becomes robust [8]. Smoothing makes model parameter values flatter by using prior knowledge on data or by combining different types of parameter estimates such as estimates on

simpler models. It is a more general technique that may include the two mentioned above. As far as language models (LMs), types of probabilistic models which we will use in Chapter 3, are concerned, Chen and Goodman empirically compared various smoothing techniques [9], and for LM-based information retrieval, Zhai and Lafferty also carried out an empirical comparison of various smoothing techniques [10]. Although these three methods are practically important, they have an obvious limitation: they cannot reveal anything that the data do not suggest. If the desired performance of the model is beyond the capability of given information, additional information is needed for the learning.

1.3. Looking for Additional Information

A straightforward solution to the information insufficiency problem is, of course, to collect more data. Sometimes gathering additional data is the only way to tackle the problem. In many situations, however, data collection is too expensive. Here are some examples.

- You need video data taken from documentary films with transcriptions. Transcriptions are supposed to indicate from which part to which part the content is boring. To provide transcriptions, human annotators have to watch hours of videos—not to enjoy them but to evaluate them.
- You suspect that a compound may exhibit some strange behavior if substances A and B exist in it at the same time. Both A and B are found frequently in the compound but they seldom co-exist. In that case, hundreds of measurements of the compounds can provide sufficient information on the statistical properties of A and B but never tell the outcomes of their interaction, which is what you want to know. Heavy repetition may be needed to obtain that information.
- You want to categorize music CDs according to genre. You also want a categorization system that will be useful for people searching for the CDs. Different people have different conceptions of music categories; therefore, to be most useful, category indexing must reflect the majority opinion. Asking whether certain categorization is relevant to somebody is often time consuming and probably stressful for the respondent.

These are only a few examples but clearly, in many situations, gathering additional data is prohibitive due to cost.

Now let us see when and where such difficulty may occur most seriously. Viewed from specific applications, examples like the ones above are often found in two important information processing tasks: classification and information retrieval (IR), and they may suffer from severe information insufficiency. When training classifiers, label information such as the indices of categories or good/bad judgments must be provided. Labels are usually created by human effort and labeling sometimes involves expert knowledge or careful investigation. Because of the use of such labeling, the problem of label scarcity occurs frequently. In IR, documents

are ranked according to the the relevance to the user's need, and this need is represented by a query. The availability of relevance information to a specific query is unlikely before the query is actually issued. Moreover, reliability of relevance information is questionable as it is derived from subjective ratings. Thus, it is common to build IR models without relying on the relevance information. In such IR models, the ranking is performed by using similarities between the query and documents based on word-occurrence information. Accordingly, complex word-word relationships must be estimated from finite examples or documents. The vocabularies of natural languages are enormous, whereas the observable patterns of their appearances are limited. Information insufficiency is, thus, evident in IR. Because it is so difficult to gather training data as exemplified above, it is important to overcome information insufficiency without a data collection cost.

When information is insufficient and we cannot gather any more data, the use of additional information source emerges as a potentially powerful approach. Although the importance of research on learning from multiple sources has been frequently raised in literature (e.g., [11],[12]), practical algorithms for specific applications have not been fully prepared. Presumably, there is no universal procedure that can successfully combine all types of additional sources in any application because information sources are *heterogeneous* and all additional sources have their own unique characteristics. We investigate the usage of available information sources in the two distinct application domains (classification and IR). As mentioned above, these two tasks suffer particularly heavily from information insufficiency. The information sources considered are unlabeled data and paired representations, respectively. We present for the two applications novel multi-source learning algorithms that break through the data heterogeneity. The following sections explain the contributions of this thesis more specifically.

1.4. Data Scarcity in Classification

The first application we consider is classification problems in which optimal class labels are automatically assigned to observations whose class labels are unknown. Generally, a classifier is trained only on labeled data consisting of observed feature values and their class labels. However, because of the high labeling cost, the quantity of labeled training data is often small. We call this information insufficiency *data scarcity*. In such a case, the model over-fits the training data and will typically generalize poorly [13]. Namely, such a trained model cannot classify unseen data well.

In this thesis, we utilize a learning scheme called *semi-supervised learning* in which both labeled and *unlabeled data* are used to increase the size of training data. In many situations, feature values are measured automatically with lower cost and are readily available. Consequently, unlabeled data are now some of the most eagerly-explored information sources to overcome the training data scarcity. Broadly speaking, unlabeled data are included in the category of incomplete data. The degree of incompleteness, however, varies. For example, a datum can be considered weakly labeled when its class label is given to a group of data rather than

Table 1.1. This table summarizes various techniques that use unlabeled data in learning classifiers when training data are scarce. Particular emphasis is on the model type, which is either designed for static data or for dynamic data. Unlabeled data can be integrated with the output of classifiers (pseudo-labels) or can be used as the class-weighted unlabeled data.

Model Type	Procedure Type	Examples of Algorithms
Static	Model Output-Based	Co-Training [20], [21]
	Within Learning Process	EM-Based [25], [26], [27]
Dynamic	Model Output-Based	Naive Labeling [22], [23]
	Within Learning Process	This Thesis

to a single datum [14]. Another type of incompleteness may occur when label information on a datum is missing though labels of surrounding data are known [15]. The unlabeled data we consider here feature absolute incompleteness.

Some theoretical and empirical results on the properties of unlabeled data have been provided [16], [17], [18], [19], and from among various types of classification problems, this thesis concerns the classification of a series of observed points. As depicted in Table 1.1, unlabeled data have been used in learning of classifiers for static data. Unlabeled sequences, however, have not.

Let us now look at the types of semi-supervised algorithms. We categorize them into two families. The first branch includes model-output level merging algorithms, in which tentatively built classifiers assign pseudo-class labels to unlabeled data. Co-training is an example. Classifiers are learned from two feature sets and from both labeled and unlabeled data [20]. Co-updating has similar properties in that it uses both additional features and unlabeled data as the additional sources [21]. The scheme uses only outputs of a model rather than the internal values of models, which could be an advantage because original fully supervised learning algorithms need not be changed. The limiting factor of co-training algorithms, however, is that they require redundant features, meaning that as well as a pool of unlabeled data, other information sources are needed. In fact, the model-output level semi-supervised algorithms that have actually been used in dynamic models are not co-training algorithms. Instead, a less powerful method, which we call the naive labeling (NL) method, has been used [22],[23]. In the NL method, only a single set of feature values is used and redundant features are not needed, although the usefulness of the method seems to be mixed. Chapter 2 discusses the NL method in detail.

Another branch of semi-supervised algorithms utilizes the assumption that class labels are random variables. By using an iterative learning algorithm called the EM algorithm [24], unlabeled data are used as class-weighted data during the learning process. EM-based semi-supervised algorithms have been derived and applied to some probabilistic models such as Gaussian mixture models [25], mixture of expert models [26], and naive Bayes models [27]. These are, however, models for static data and do not directly address sequences. In this thesis,

we consider the use of unlabeled sequences in learning hidden Markov models (HMMs). HMMs are the most frequently used probabilistic models for sequences. Application areas of HMMs include speech recognition, gesture recognition, natural language processing, bioinformatics, and many more. We consider the construction of semi-supervised algorithms for HMMs valuable due to their practical importance.

Before explaining our contributions, a few remarks on the definition of unlabeled data should be made. On one hand, unlabeled data for the static models are defined uniquely. On the other hand, they follow two definitions of unlabeled sequences. The first one is the straightforward extension of unlabeled static data: one sequence corresponds to one label. The second one assumes partial labeling: one sequence corresponds to a series of labels. Of the two definitions, we focus on the first basic case.

The contribution of this thesis to the data scarcity problem in learning HMMs is threefold:

- We introduce a mixture of HMMs to utilize unlabeled sequences.
- We formally show the semi-supervised EM algorithm for the above HMMs that uses unlabeled sequences. There is already a method that labels unlabeled sequences then uses them as if they are labeled data. However, we first propose a method that uses both labeled and unlabeled sequences "simultaneously."
- We experimentally show that our method can improve classification performances for the two types of real data. We also show that our method is superior to the above-mentioned conventional method.

1.5. Data Sparseness in Information Retrieval

We consider IR problems as the second application. In many IR tasks, the vocabulary is very large. In contrast, the variety of words used in a single document is very limited. For example, the variety of words contained in a document is usually less than 1% of the whole vocabulary, which means the information on word usage obtainable from a document is very sparse compared to the total vocabulary. We call this type of insufficiency *data sparseness*. As Katz claimed, the situation deteriorates if we want to use some higher structural information among words such as compound words [28], even if we have a sufficient amount of data entries, such sparseness of information is still a serious problem in IR. The relationships between words play a crucial role in probabilistic model-based IR systems but such information is difficult to obtain. To be successful, learning algorithms usually require additional sources that fill in the sparseness of training data.

From among various types of IR problems, this thesis deals with the cross-media IR, in which queries and corresponding target data belong to different media. In the history of IR, textual data, which are symbolic data, have been the main concerns of researchers. These days, however, large collections of signal data such as images are very common and effective retrieval techniques for these data are needed. Signal data can be retrieved by comparing one

Table 1.2. This table summarizes various techniques that use additional information in learning IR systems when training data are sparse. Particular emphasis is on the type of IR, either mono-media or cross-media. Additional information can be either pre-structured knowledge or low-level signals.

Task Type	Information Type	Examples of Sources
Mono-media IR	Knowledge Level	Thesaurus [29], [30]
	Feature Level	Mixed Data [31], [32]
Cross-media IR	Knowledge Level	Thesaurus [33], [34]
	Feature Level	This Thesis

example of data to the other data in the collection. Unfortunately, such sample images are not always available nor manipulable. The easiest and most effective way to issue a query to retrieve such signal data is to use natural language. In this case, the IR task becomes a cross-media one, using textual queries and visual documents. Therefore, cross-media IR is quite an important research topic in modern IR that should be explored further. When textual annotations explaining the contents of images are provided, such cross-media image retrieval can be approached by adopting conventional textual IR techniques on annotations. The problem of data sparseness, however, becomes more serious in such annotation-based cross-media IR. Compared to ordinary texts, annotations contain fewer words, and the learning of semantic relationships between words becomes more difficult.

In mono-media IR where queries and documents are represented on the same medium, additional information has been used in various ways (See Table 1.2). Additional information can also be introduced as knowledge; a thesaurus is a typical example and has been used in textual IR [29], [30]. Semantic relationships between words are manually defined within it. Additional information can also be supplied as feature values. An example in textual IR is the use of multilingual text databases. The learning of a probabilistic model for a minor language based on the documents in the WWW has been improved by modeling other languages as well using documents obtained from the WWW [31]. Another example uses both textual and visual data in visual IR [32]. These researches on multi-source learning are still in their infancy but are gradually gaining attentions.

In contrast to mono-media IR, cross-media IR itself has not been studied extensively. Consequently, relatively fewer researches on using additional sources have been carried out (See Table 1.2). As for already-structured information sources, as well as textual IR, thesauri have been used for cross-media visual IR [33], [34], while as for the unorganized feature level, incorporation techniques still seem to be unexplored. We enter this field with the idea that annotations and images are correlated. We regard images as the paired data of annotations and propose a method for estimating similarities among annotation words by using similarity between images. It can be said that, in our algorithm, the paired image data are used to interpolate the sparseness of textual annotations. We consider this a new way to use heterogeneous information

sources in cross-media IR.

The contribution of this thesis to the problem of data sparseness in cross-media IR is three-fold:

- We introduce a probabilistic model that performs query-by-text cross-media IR.
- We propose an algorithm that utilizes paired data and mitigates the data sparseness problem in learning of the above model.
- We experimentally show that the proposed algorithm can improve the IR performances for the photo image dataset when the amount of annotations is limited.

1.6. Summary of Remaining Chapters

The remaining chapters are organized as follows. In Chapter 2, we present the method for exploiting unlabeled sequences in learning HMMs and experimentally evaluate it using sign-language data and speech phoneme data. Chapter 3 explains our method to incorporate paired data in learning a model for cross-media image retrieval. Experimental results on photo image retrieval are also shown. Finally, Chapter 4 concludes the thesis.

Chapter 2

Unlabeled Sequences in Hidden Markov Models

2.1. Introduction to This Chapter

One major problem in designing classifiers is the scarcity of training data. Usually, a classifier is trained on pairs of observed feature vectors and their class labels. Such a framework is called supervised learning. In most cases, class labels are manually assigned by experts. Therefore, it is expensive and time consuming to collect large amounts of labeled data. Because of this labeling cost, data is often scarce in practice. Consequently, the designed classifier becomes unreliable and its generalization performance becomes poor [35], especially in nonlinear models.

To overcome this problem in supervised learning, a new learning scheme called *semi-supervised learning* has been proposed in which unlabeled data are also used to train classifiers. Since unlabeled data can be easily collected without labeling efforts, semi-supervised learning has attracted classifier designers, and has been studied in various applications for both static data [36], [37], [25], [26], [27], and sequential data [22], [23]. It was reported that the classifiers learned from both labeled and unlabeled data could achieve better classification performance than those learned from small amounts of labeled data.

In this paper, we focus on semi-supervised learning for hidden Markov models (HMMs). HMMs are stochastic state transition models that have been extensively used in two types of applications. The first one is concerned with classification of sequences in speech recognition (e.g. [38]), in gesture recognition (e.g. [39]), in computational biology (e.g. [40]), and so on. In these tasks, given a sequence, HMMs assign a class label to the entire sequence. The second type deals with the determination of state sequences given observation sequences. Examples of this type of applications include part of speech tagging in natural language processing (e.g. [41]) and named entity extraction in information extraction (e.g. [42]). In the second type of applications, the term "labeled data" means the observed sequences with the state sequence information

associated with them, while the term "unlabeled data" means the sequences without the state sequence information [43], [44], [45]. That is, the second one is concerned with "partially hidden data" which are not "unlabeled data" in the sense used for the semi-supervised learning for static data. Such "partially hidden data" of the second application type can be processed by the standard learning framework of HMMs and in this paper, we investigate the "unlabeled data" in the first type of application, the classification of sequences.

For the the first applications, a simple semi-supervised learning scheme has been used, which we refer to as the naive labeling (NL) approach [22], [23]. With the NL approach, HMMs are first trained solely on given labeled data. Then, pseudo class labels are *deterministically* assigned to unlabeled data by classifying them using the trained HMMs. The HMMs are retrained with these newly labeled data.

The NL approach appears to be a method for classifier adaptation under the assumption that the initial model is to some extent reliable. In the above two studies where the quantities of initial labeled data were relatively large, the NL approach could improve the HMMs. This applies to the case when training HMMs used in speech recognition systems for adaptation. However, when trained on small amounts of labeled data, initial models become unreliable and therefore the pseudo labels also become unreliable. As a result, the addition of unlabeled data with such unreliable labels may not improve the generalization performance of the HMM.

To overcome this problem associated with the NL approach, in this paper, we present a new semi-supervised learning approach that can use unlabeled data for training HMMs more effectively than the NL approach. In our approach, as with [37], [25], [26], [27], the class labels are treated as missing information and the pseudo class labels are *probabilistically* assigned to unlabeled data so that the joint likelihood function for both labeled and unlabeled data is maximized. To handle unlabeled data, we introduce extended tied-mixture HMMs (ETM-HMMs) as a mixture of tied-mixture HMMs (TM-HMMs) [46], [47]. For training ETM-HMMs, we derive an extended Baum-Welch (EBW) algorithm. Unlike the NL approach, the proposed algorithm theoretically guarantees convergence to a local maximum of the likelihood.

The EBW algorithm can be regarded as an extension of the conventional labeling approach for static data based on the EM algorithm [24] to the one for sequential data. Although the usefulness of static unlabeled data has been claimed, the usefulness of sequential unlabeled data in such approach has not been shown. In the present paper, we formally explain the EBW algorithm and empirically compare it with the NL approach.

The rest of the chapter is organized as follows. After the formal definition of labeled and unlabeled data in Section 2.2, the conventional NL approach is explained in Section 2.3. Section 2.4 briefly reviews TM-HMMs and introduces ETM-HMMs. Next, the EBW algorithm is presented in Section 2.5. Section 2.6 provides some experimental results using gesture and speech data in which the effect of unlabeled data in our method is evaluated and compared with the NL approach. Section 2.7 concludes this chapter.

2.2. Labeled and Unlabeled Data

Let $X_n = \langle \mathbf{x}_{n_1}, \mathbf{x}_{n_2}, ..., \mathbf{x}_{n_t}, ..., \mathbf{x}_{n_{T_n}} \rangle$ be the nth observation sequence of d-dimensional feature vectors, where $\mathbf{x}_{n_t} \in \mathcal{R}^d$ is the tth feature vector in X_n and T_n is the length of the sequence X_n. Let y_n be a class label corresponding to X_n. $y_n \in \{1, ..., y, ..., Y\}$ where Y is the number of classes. Thus, a labeled datum is (X_n, y_n) and an unlabeled datum is X_n. Let $\mathcal{D}_l$ be a labeled data set and $\mathcal{D}_u$ be an unlabeled data set. It is assumed that we have $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$. In addition, we assume that data are mutually independent.

In the above definition, a single label is assigned to each of the observed sequences. Many types of sequence meet the above definition. For example, physiological sequences such as brain waves, biological sequences such as gene expression profiles, and economic time series data such as the trend of unemployment. For some types of sequence, on the other hand, another definition of labeled sequences is sometimes used: an observed sequence corresponds to several concatenated labels. For example, continuous speech recognition systems that regard each phoneme as a class deal with spoken sentences as such sequences. For these settings, however, together with segmentation algorithms such as [48], the concatenated sequences may be dealt with as the basic sequences defined above. In this paper, with the aim of evaluating the proposed algorithm, we consider only the basic sequences where there is a one-to-one correspondence between a sequence of feature vectors and a class label.

2.3. Naive Labeling Approach

First, we review the conventional NL approach that utilizes unlabeled data straightforwardly. Assume that relatively small amounts of data have been manually labeled and vast amounts of unlabeled data are accessible. In the NL approach, using the hand-labeled data, a partially correct initial model is trained. The remaining unlabeled data are labeled based on the initial model. Once unlabeled data have been given pseudo labels, they can be regarded as labeled data. Then, the model can be retrained by using conventional supervised learning algorithms.

Let $\mathcal{D}_l'$ be a pseudo-labeled data set whose labels are generated by the initial model. Then, the NL approach can be summarized as follows:

Step 1: Initialization

1-1. Set $\mathcal{D} \leftarrow \mathcal{D}_l$.

1-2. Train a model (classifier) using $\mathcal{D}$.

Step 2: Retraining

Repeat the following several times:

2-1. Based on the current model, assign a pseudo label to each datum in $\mathcal{D}_u$ and generate $\mathcal{D}_l'$.

2-2. Set $\mathcal{D} \leftarrow \mathcal{D}_l \cup \mathcal{D}_l^{'}$.

2-3. Retrain the model using $\mathcal{D}$.

The above algorithm gives the most general form of the NL approach, However, since the NL approach has been developed independently for various applications, some variants and extensions exist. For example, in [36], $\mathcal{D}_l^{'}$ was used and $\mathcal{D}_l$ was not used in retraining, and in [22], step 2 was executed just once. However, such differences do not seem to be essential. Therefore, in this paper, we use the general algorithm given above.

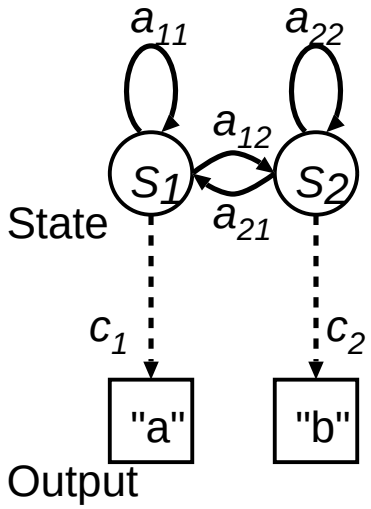
Although the NL approach has been reported to be effective in practice, it has two fundamental drawbacks. First, the convergence of step 2 is not guaranteed. Therefore, should the retraining procedure not converge, we should stop the retraining cycle based on some heuristic criterion such as the maximum number of retraining cycles. Second, when the initial model is unreliable, the unlabeled data cannot be effectively used. Since $\mathcal{D}_l$ in step 1-1 is often small, the initial model may be poorly trained; thus, a substantial percentage of pseudo labels assigned by such models may be wrong. If $\mathcal{D}_l^{'}$ contains many erroneous data, their addition may deteriorate the performance of the classifier.

Confidence measures for labeling have been introduced to cope with the second problem [22], [23] so that unreliable pseudo labeled data whose confidence measures are below a certain threshold are not included in $\mathcal{D}$. These confidence measures are defined for individual applications based on domain knowledge and have been reported to be beneficial in improving classification. Such measures are, however, not always available or effective. That is, the success of the NL approach basically depends on the quality of the initial model.

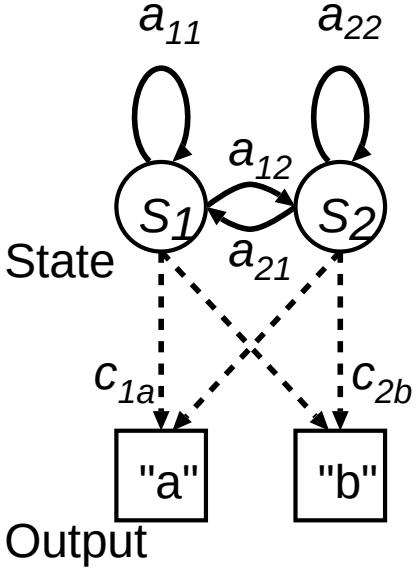
2.4. TM-HMMs and ETM-HMMs

2.4.1 Proposed Algorithm and Model Structure

To overcome the problems associated with the NL approach, we propose a new algorithm which uses unlabeled data directly without explicit labeling. By so doing, we can use both labeled and unlabeled data simultaneously and expect a better initial estimate of the model parameters based on the larger amount of training data. Such methods have been already presented for static models [37], [25], [26], [27]. However, the structure of conventional HMMs prevents the direct use of these methods. The static models used in the above researches include all classes in a single model and unlabeled data can be used in those models. In contrast, HMMs are constructed for each class and all training data must be allocated to the classes before learning. Therefore, unlabeled data cannot be used unless pseudo-labels are given by a method such as the NL approach. In Section 2.4.2, we briefly review the definition of HMMs. In Section 2.4.3, to clarify why unlabeled data cannot be used in conventional HMMs, we detail the model structure of conventional HMMs especially tied-mixture HMMs (TM-HMMs). In Section 2.4.4, as an extension of TM-HMMs, we introduce a model structure named extended tied-mixture



(a)MM



(b)HMM

Figure 2.1. This figure shows a two-state (S_1 and S_2) example of (a) a Markov model (MM) and (b) a hidden Markov model (HMM). The circles represent HMM states and the solid arrows represent state transitions. The squares denote output space of symbols. In (a), output symbol "a" always comes from state S_1 and "b" from S_2. In contrast, in (b), such the relationship between states and outputs are probabilistic (hidden).

HMMs (ETM-HMMs) that can handle unlabeled data. Actually, our ETM-HMMs is not the only way to use unlabeled sequences in HMMs; another model structure is also possible (See Appendix A).

2.4.2 HMMs

An HMM consists of several states and the probabilistic transitions between them. Its transition from one state to another depends only on the current state. Such a property is said to be Markovian. If the outputs from Markovian models correspond to their state one-to-one, such models are called Markov models (MMs). Hidden Markov models are different. When a sequence of output values from an HMM $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t\}$ is observed, it is considered to be generated from state sequence of HMMs. Such a sequence is not observable, in contrast to the observable output sequences, which is the reason for the name "hidden." Figure 2.1 shows simple a two-state example of an MM and an HMM. The basic HMMs output symbolic values. Continuous (vector) values can be dealt by with an extension of HMMs called continuous HMMs. Continuous HMMs output a continuous value from each state according to the distribution of a mixture of Gaussians associated with them. Formal definitions of HMMs will be given in the next section for a variant of HMMs called TM-HMMs.

2.4.3 TM-HMMs

In a TM-HMM shown in Fig. 2.2(a), each state has a mixture of Gaussians with shared underlying Gaussian components over all classes, but different mixing parameters. TM-HMMs are frequently used because they can reduce the number of model parameters without losing flexibility [46], [47].

Let TM-HMM(y) be a TM-HMM of class y. Let U^y be the number of states in TM-HMM(y) and K be the number of Gaussian components in the feature space. Let $s_t \in \{1, ..., i, ..., j, ..., U^y\}$ be the index of the state at time t. [1] Let $m_t \in \{1, ..., k, ..., K\}$ be the index of the component at time t. Let $\Theta_y = \{\pi_i^y, a_{ij}^y, c_{jk}^y, \boldsymbol{\mu}_k, \Sigma_k\}$ be the set of parameters for TM-HMM(y). The definitions of parameters in Θ_y are listed below.

- Initial state probabilities for $1 \leq s_1 \leq U^y$:

$$\begin{aligned} \pi_i^y &= P(s_1 = i|y), \\ \text{where} \quad & \pi_i^y \geq 0 \text{ and } \sum_i \pi_i^y = 1. \end{aligned} \tag{2.1}$$

- Transition probabilities for $1 \leq s_t, s_{t+1} \leq U^y$:

$$\begin{aligned} a_{ij}^y &= P(s_{t+1} = j|s_t = i, y), \\ \text{where} \quad & a_{ij}^y \geq 0 \text{ and } \sum_j a_{ij}^y = 1. \end{aligned} \tag{2.2}$$

[1] Since indices i and j represent a state of the HMM for a particular class (y), they should be written as i^y and j^y. However, for simplicity of notation, we omit the superscript y.

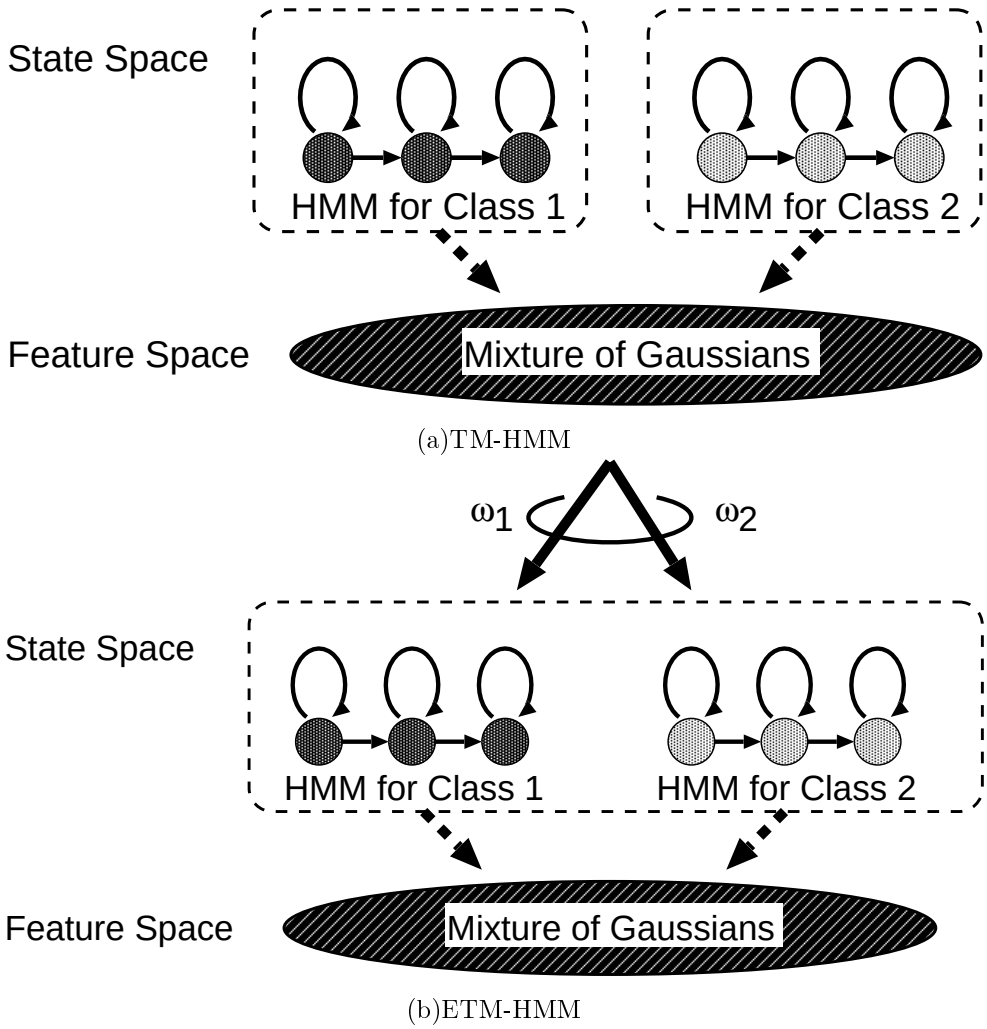


Figure 2.2. This figure shows a two-class example of (a) a Tied-Mixture HMM (TM-HMM) and (b) an Extended Tied-Mixture HMM (ETM-HMM). The circles represent HMM states and the solid arrows represent state transitions. The black ovals denote feature spaces represented by a mixture of Gaussians. In (b), ω_1 and ω_2 denote class priors for class 1 and class 2, respectively.

- Mixture coefficients for $1 \leq s_t \leq U^y, \quad 1 \leq m_t \leq K$:

$$\begin{aligned} c^y_{jk} &= P(m_t = k | s_t = j, y), \\ \text{where} \qquad & c^y_{jk} \geq 0 \text{ and } \sum_k c^y_{jk} = 1. \end{aligned} \tag{2.3}$$

- Gaussian parameters (mean vectors and covariance matrices) for $1 \leq m_t \leq K$:

$$\begin{aligned} &\boldsymbol{\mu}_k \text{ and } \Sigma_k, \\ &\text{where } m_t = k. \end{aligned} \tag{2.4}$$

Note that since all Gaussians are common to all states and classes, $\boldsymbol{\mu}_k$ and Σ_k depend neither on i, j nor on y.

Let $S_n = \{s_t | \ t = 1, ..., T_n\}$ be the sequence of states, $M_n = \{m_t | \ t = 1, ..., T_n\}$ be the sequence of Gaussian components both of which correspond to X_n. In TM-HMM(y), X_n is observable and S_n and M_n are unobservable; hence, they are called *hidden* variables. According to the definition of Θ_y, when $s_1 = h, s_t = i, s_{t+1} = j, m_t = k$, the complete data likelihood of TM-HMM(y) is given by:

$$\begin{aligned} & p(X_n, S_n, M_n | \Theta_y) \\ &= \pi^y_h \prod_{t=1}^{T_n - 1} a^y_{ij} \prod_{t=1}^{T_n} c^y_{ik} \mathcal{N}(\mathbf{x}_{n_t} | \ \boldsymbol{\mu}_k, \Sigma_k). \end{aligned} \tag{2.5}$$

In TM-HMMs, as shown in Fig. 2.3(a), the feature space is tied over classes and any feature vector can be placed there. In contrast, since state spaces are defined separately for each TM-HMM(y), unlabeled data without class labels cannot be placed in state spaces. Therefore, TM-HMMs cannot use unlabeled data directly.

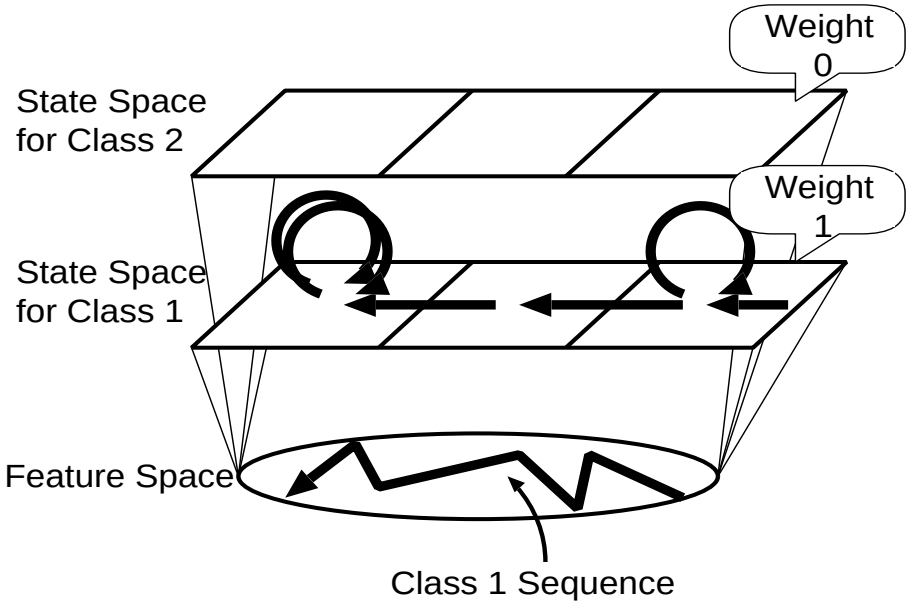
2.4.4 ETM-HMMs

To deal with unlabeled data in state space, we use another model structure named an ETM-HMM. Let $P(y) = \omega_y$ be a class prior. Then, an ETM-HMM can be defined by:

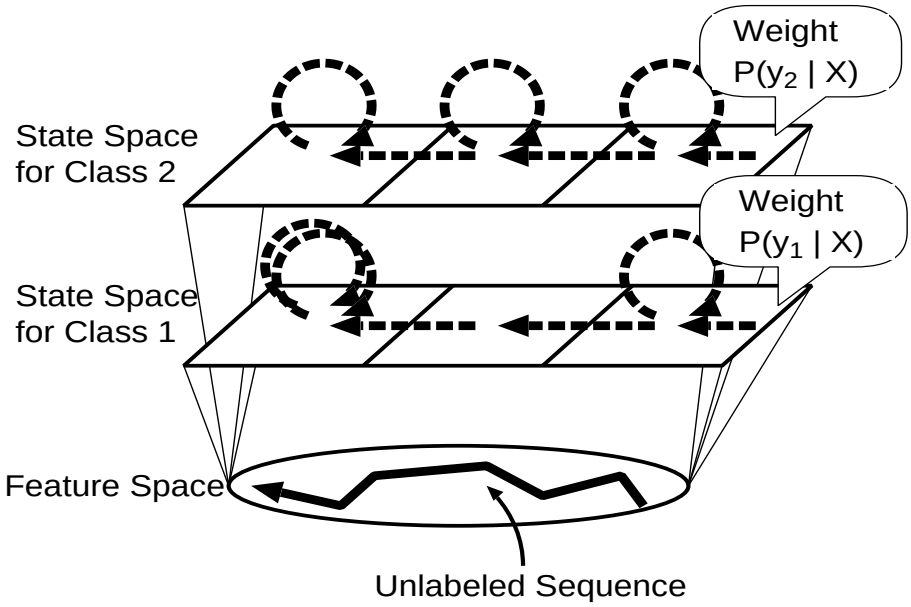
$$\text{ETM-HMM} = \textstyle\sum_{y=1}^{Y} \omega_y \text{TM-HMM}(y). \tag{2.6}$$

That is, an ETM-HMM is defined as a mixture of TM-HMMs of different classes as shown in Fig. 2.2(b). In an ETM-HMM, as well as a TM-HMM, a feature space represented by a mixture of Gaussians is tied over all classes. As shown in Fig. 2.3(b), in ETM-HMMs, unlabeled data can be located in multiple state spaces with probabilistic weights.

The hidden variables of an ETM-HMM for labeled data are that for a TM-HMM, and for unlabeled data, class label y_n is also a hidden variable. A set of parameters for an ETM-HMM $\Theta = \{\omega_y, \Theta_y | y = 1, ..., Y\}$, where Θ_y has been defined for TM-HMMs in Section 2.4.3. When



(a) Labeled Data (TM-HMMs or ETM-HMMs)



(b) Unlabeled Data (ETM-HMMs)

Figure 2.3. Allocations of (a) labeled and (b) unlabeled sequences in the state spaces of two classes. Labeled sequences can be located in either TM-HMMs or an ETM-HMM. By contrast, unlabeled sequences can be located only in an ETM-HMM.

$s_1 = h, s_t = i, s_{t+1} = j$, and $m_t = k$, the complete data likelihood for an ETM-HMM is given by:

$$\begin{aligned} & p(X_n, y_n, S_n, M_n|\Theta) \\ = \ & \omega_y \pi_h^y \prod_{t=1}^{T_n-1} a_{ij}^y \prod_{t=1}^{T_n} c_{ik}^y \mathcal{N}(\mathbf{x}_{n_t}|\ \boldsymbol{\mu}_k, \Sigma_k). \end{aligned} \tag{2.7}$$

Equation (2.7) differs from (2.5) in that class prior ω_y is introduced and class labels are regarded as random variables.

2.5. Extended Baum-Welch Algorithm

2.5.1 Q-function for Mixed Data

This section describes the learning algorithm for the ETM-HMMs, named the extended Baum-Welch (EBW) algorithm. The EBW algorithm is an extension of the Baum-Welch (BW) algorithm [49], which is widely used to train HMMs. The BW algorithm can be regarded as an application of the expectation-maximization (EM) algorithm [24] to HMMs. The EM algorithm is an iterative procedure for computing maximum likelihood estimates from incomplete data. It alternates two steps: E-step computes the expected complete data log-likelihood called Q-function, and M-step maximizes the Q-function with respect to unknown parameters. Since the ETM-HMM learns from both labeled and unlabeled data, the Q-function for the conventional HMM needs to be redefined.

First, we derive the Q-function for labeled and unlabeled mixed data in a general form. Let $\mathcal{Z}_l$ and $\mathcal{Z}_u$ be sets of hidden variables that correspond to $\mathcal{D}_l$ and $\mathcal{D}_u$, respectively. Then, a set of hidden variables $\mathcal{Z} = \mathcal{Z}_l \cup \mathcal{Z}_u$ corresponds to $\mathcal{D}$. Each datum is either labeled ($d_l \in \mathcal{D}_l$ and $z_l \in \mathcal{Z}_l$) or unlabeled ($d_u \in \mathcal{D}_u$ and $z_u \in \mathcal{Z}_u$). Let θ be a set of unknown model parameters. Assuming that data are independently and identically distributed (i.i.d.), we can decompose complete data likelihood into the complete data likelihood for labeled data and that for unlabeled data:

$$p(\mathcal{D}, \mathcal{Z}|\theta) = p(\mathcal{D}_l, \mathcal{Z}_l|\theta) \cdot p(\mathcal{D}_u, \mathcal{Z}_u|\theta). \tag{2.8}$$

Similarly, since $p(\mathcal{D}) = p(\mathcal{D}_l) \cdot p(\mathcal{D}_u)$ holds based on the assumption of the independence between data, the distribution of posterior probabilities for the hidden variable $\mathcal{Z}$ given current parameter estimates θ^{old} can be decomposed as shown below:

$$\begin{aligned} P(\mathcal{Z}|\mathcal{D}, \theta^{\text{old}}) &= \frac{p(\mathcal{D}, \mathcal{Z}|\theta^{\text{old}})}{p(\mathcal{D})} \\ &= \frac{p(d_{l_1}, z_{l_1}, d_{l_2}, z_{l_2}, ..., d_{u_1}, z_{u_1}, d_{u_2}, z_{u_2}, ...|\theta^{\text{old}})}{p(d_{l_1}, d_{l_2}, ..., d_{u_1}, d_{u_2}, ...)} \\ &= \frac{p(\mathcal{D}_l, \mathcal{Z}_l|\theta^{\text{old}})}{p(\mathcal{D}_l)} \cdot \frac{p(\mathcal{D}_u, \mathcal{Z}_u|\theta^{\text{old}})}{p(\mathcal{D}_u)} \\ &\equiv P(\mathcal{Z}_l|\mathcal{D}_l, \theta^{\text{old}}) \cdot P(\mathcal{Z}_u|\mathcal{D}_u, \theta^{\text{old}}). \end{aligned} \tag{2.9}$$

By definition, the general formulation of the Q-function is given by:

$$\begin{aligned} Q(\theta|\theta^{\text{old}}) &= \mathrm{E}\left[\log p(\mathcal{D}, \mathcal{Z}|\theta)|\ \mathcal{D}, \theta^{\text{old}}\right] \\ &= \sum_{\mathcal{Z}} P(\mathcal{Z}|\mathcal{D}, \theta^{\text{old}}) \log p(\mathcal{D}, \mathcal{Z}|\theta). \end{aligned} \tag{2.10}$$

Therefore, by substituting (2.8) and (2.9) into (2.10), the Q-function for the mixed data can be obtained:

$$Q(\theta|\theta^{\text{old}}) = Q_l(\theta|\theta^{\text{old}}) + Q_u(\theta|\theta^{\text{old}}), \tag{2.11}$$

where

$$Q_l(\theta|\theta^{\text{old}}) = \mathrm{E}\left[\log p(\mathcal{D}_l, \mathcal{Z}_l|\theta)|\ \mathcal{D}_l, \theta^{\text{old}}\right]$$

and

$$Q_u(\theta|\theta^{\text{old}}) = \mathrm{E}\left[\log p(\mathcal{D}_u, \mathcal{Z}_u|\theta)|\ \mathcal{D}_u, \theta^{\text{old}}\right].$$

To sum up, the Q-function for the mixed data is the direct sum of the Q-functions for labeled and unlabeled data.

2.5.2 E-step: Calculation of Q-function

Applying (2.11) to ETM-HMMs, we derive the Q-function for the EBW algorithm. Let N_l be the number of labeled sequential data and $\mathcal{I}_y$ be the set of data indices $\{n|\ y_n = y\}$. Labeled data are $\{(X_n, y_n) \in \mathcal{D}_l\}$ and the corresponding hidden variables are $\{(S_n, M_n) \in \mathcal{Z}_l\}$. By taking the conditional expectation of the complete log-likelihood (log of (2.7)) over the hidden variables given the data and the current estimates of parameters Θ^{old}, the Q-function for labeled data, Q_l, is derived as follows:

$$\begin{aligned} & Q_l(\Theta|\Theta^{\text{old}}) \\ = & \sum_{n=1}^{N_l} \mathrm{E}\left[\log p(X_n, y_n, S_n, M_n|\Theta)|\ X_n, y_n, \Theta^{\text{old}}\right] \\ = & \sum_{y=1}^{Y} \sum_{n \in \mathcal{I}_y} \Big\{ \log \omega_y \\ + & \sum_j \gamma_{n_0}(j) \log \pi_j^y \\ + & \sum_{i,j} \sum_{t=1}^{T_n - 1} \gamma_{n_t}(i,j) \log a_{ij}^y \\ + & \sum_{j,k} \sum_{t=1}^{T_n} \zeta_{n_t}(j,k) \log c_{ik}^y \\ + & \sum_k \sum_{t=1}^{T_n} \kappa_{n_t}(k) \log \mathcal{N}(\mathbf{x}_{n_t}|\boldsymbol{\mu}_k, \Sigma_k) \Big\}. \end{aligned} \tag{2.12}$$

Next, the posterior probabilities of hidden variables defined as (2.9) are specified. In (2.12), γ, ζ and κ represent transition posteriors, staying and emission posteriors, and emission posteriors given below for $t \geq 1$:

$$\gamma_{n_t}(i,j) = P(s_t = i,\ s_{t+1} = j|\ X_n,\ y_n,\ \Theta^{\text{old}}), \tag{2.13}$$

$$\zeta_{n_t}(j,k) = P(s_t = j,\ m_t = k|\ X_n,\ y_n,\ \Theta^{\text{old}}), \tag{2.14}$$

$$\kappa_{n_t}(k) = P(m_t = k|\ X_n,\ y_n,\ \Theta^{\text{old}}). \tag{2.15}$$

Note that in (2.13), $\gamma_{n_0}(j) = P(s_1 = j|\ X_n,\ y_n,\ \Theta^{\text{old}})$.

Let N_u be the number of unlabeled sequential data. Unlabeled data are $\{X_n \in \mathcal{D}_u\}$ and the corresponding hidden variables are $\{(y_n, S_n, M_n) \in \mathcal{Z}_u\}$. By taking the conditional expectation of the complete log-likelihood (log of (2.7)) over the hidden variables given the data and the current estimates of parameters Θ^{old}, the Q-function for unlabeled data, Q_u, is derived as follows:

$$\begin{aligned}
& Q_u(\Theta|\Theta^{\text{old}}) \\
= & \sum_{n=1}^{N_u} \mathrm{E}\left[\log p(X_n, y_n, S_n, M_n|\Theta)|\ X_n, \Theta^{\text{old}}\right] \\
= & \sum_{n=1}^{N_u}\sum_{y=1}^{Y}\Big\{ P(y|X_n, \Theta^{\text{old}}) \log \omega_y \\
+ & \sum_j \lambda_{n_0}(y,j) \log \pi_j^y \\
+ & \sum_{i,j}\sum_{t=1}^{T_n-1} \lambda_{n_t}(y,i,j) \log a_{ij}^y \\
+ & \sum_{j,k}\sum_{t=1}^{T_n} \eta_{n_t}(y,j,k) \log c_{ik}^y \\
+ & \sum_{k}\sum_{t=1}^{T_n} \xi_{n_t}(y,k) \log \mathcal{N}(\mathbf{x}_{n_t}|\boldsymbol{\mu}_k, \Sigma_k)\Big\}.
\end{aligned} \tag{2.16}$$

In (2.16), λ, η, and ξ represent transition posteriors, staying and emission posteriors, and emission posteriors given below for $t \geq 1$:

$$\lambda_{n_t}(y,i,j) = P(y_n = y,\ s_t = i, s_{t+1} = j|\ X_n,\ \Theta^{\text{old}}), \tag{2.17}$$

$$\eta_{n_t}(y,j,k) = P(y_n = y,\ s_t = j, m_t = k|\ X_n,\ \Theta^{\text{old}}), \tag{2.18}$$

$$\xi_{n_t}(y,k) = P(y_n = y,\ m_t = k|\ X_n,\ \Theta^{\text{old}}). \tag{2.19}$$

Note that in (2.17), $\lambda_{n_0}(y,j) = P(y_n = y, s_1 = j|\ X_n,\ \Theta^{\text{old}})$. Equations (2.17)–(2.19) correspond to (2.13)–(2.15), respectively but differ in that the class label y is regarded as the value of the random variable. For unlabeled data, the class posterior given below should also be calculated.

$$P(y|\ X_n,\ \Theta^{\text{old}}). \tag{2.20}$$

Either for labeled data or for unlabeled data, the forward-backward algorithm [50] efficiently calculates the above posteriors. For unlabeled data, however, due to computational problems, a modified scaling technique needs to be applied in practice (See Appendix B).

The Q-function of the ETM-HMM is given by the sum of (2.12) and (2.16). It is different from that of the TM-HMM in the following two respects. First, Q_u does not exist in the Q-function for TM-HMMs since TM-HMMs cannot handle unlabeled data. Second, the term for ω_y does not exist in Q_l for TM-HMMs in which class priors are not taken into account.

2.5.3 M-step: Parameter Re-estimation

In the M-step, the Q-function that was derived in Section 2.5.2 is maximized with respect to each model parameter. For example, the re-estimation formula for class prior ω_y can be obtained by maximizing the objective function $J = Q(\Theta|\Theta^{\text{old}}) + \tau(\sum_{y=1}^{Y} \omega_y - 1)$ with the constraint $\sum_{y=1}^{Y} \omega_y = 1$ where τ is a Lagrange multiplier. By solving the two equations, $\partial J/\partial \omega_y = 0$ and $\partial J/\partial \tau = 0$, the following re-estimation formula is obtained:

$$\hat{\omega}_y = \frac{N_y + \sum_{n=1}^{N_u} P(y|X_n, \Theta^{\text{old}})}{N_l + N_u}, \tag{2.21}$$

where $\hat{\omega}_y$ denotes newly estimated ω_y and N_y represents the number of labeled data belonging to class y.

In a similar manner, the re-estimation formulae for transition probabilities and mixture coefficients are obtained as follows: [2]

$$\hat{a}_{ij}^{y} = \frac{\sum_{n \in \mathcal{I}_y} \sum_{t=1}^{T_n - 1} \gamma_{n_t}^{y}(i,j) + \sum_{n=1}^{N_u} \sum_{t=1}^{T_n - 1} \lambda_{n_t}(y,i,j)}{\sum_{n \in \mathcal{I}_y} \sum_{t=1}^{T_n - 1} \sum_{j} \gamma_{n_t}^{y}(i,j) + \sum_{n=1}^{N_u} \sum_{t=1}^{T_n - 1} \sum_{j} \lambda_{n_t}(y,i,j)}, \tag{2.22}$$

$$\hat{c}_{jk}^{y} = \frac{\sum_{n \in \mathcal{I}_y} \sum_{t=1}^{T_n} \zeta_{n_t}^{y}(j,k) + \sum_{n=1}^{N_u} \sum_{t=1}^{T_n} \eta_{n_t}(y,j,k)}{\sum_{n \in \mathcal{I}_y} \sum_{t=1}^{T_n} \sum_{k} \zeta_{n_t}^{y}(j,k) + \sum_{n=1}^{N_u} \sum_{t=1}^{T_n} \sum_{k} \eta_{n_t}(y,j,k)}. \tag{2.23}$$

[2] The re-estimation formula for π_i^y is not given here because we used the left-to-right models in which π_i^y is unchanged by re-estimation.

The re-estimation formulae for mixture components, which are Gaussians, are given as follows:

$$\hat{\boldsymbol{\mu}}_k = \frac{\sum_{y=1}^{Y}\left\{\sum_{n\in\mathcal{I}_y}\sum_{t=1}^{T_n}\kappa_{n_t}^{y}(k)\mathbf{x}_{n_t}+\sum_{n=1}^{N_u}\sum_{t=1}^{T_n}\xi_{n_t}(y,k)\mathbf{x}_{n_t}\right\}}{\sum_{y=1}^{Y}\left\{\sum_{n\in\mathcal{I}_y}\sum_{t=1}^{T_n}\kappa_{n_t}^{y}(k)+\sum_{n=1}^{N_u}\sum_{t=1}^{T_n}\xi_{n_t}(y,k)\right\}}, \tag{2.24}$$

$$\hat{\Sigma}_k = \frac{\sum_{y=1}^{Y}\left\{\sum_{n\in\mathcal{I}_y}\sum_{t=1}^{T_n}\kappa_{n_t}^{y}(k)\mathbf{v}_{kt}+\sum_{n=1}^{N_u}\sum_{t=1}^{T_n}\xi_{n_t}(y,k)\mathbf{v}_{kt}\right\}}{\sum_{y=1}^{Y}\left\{\sum_{n\in\mathcal{I}_y}\sum_{t=1}^{T_n}\kappa_{n_t}^{y}(k)+\sum_{n=1}^{N_u}\sum_{t=1}^{T_n}\xi_{n_t}(y,k)\right\}}, \tag{2.25}$$

where $\mathbf{v}_{kt}=(\mathbf{x}_{n_t}-\boldsymbol{\mu}_k)(\mathbf{x}_{n_t}-\boldsymbol{\mu}_k)^t$ (the superscript t denotes transpose).

At each stage of parameter re-estimation, the increase in the likelihood is guaranteed. We stop the EM cycle when the change in the log-likelihood value is below a specified threshold. Unfortunately, when mixed data are used, accurate calculation of the log-likelihood is computationally difficult; therefore, we use another measure as a convergence criterion (See Appendix C).

When only labeled data are used, the re-estimation formula for the class priors becomes $\hat{\omega}_y = N_y/N_l$. It is a constant defined by the number of training data for each class. Other parameter re-estimation formulae become those constituted from the first term of both numerators and denominators. Such formulae are the same as those in the BW algorithm. Therefore, if the class priors are the same among classes, in other words, if we assume all classes have the same number of data, the EBW algorithm without unlabeled data is reduced to the BW algorithm. That is, the BW algorithm can be viewed as a special case of the EBW algorithm.

2.5.4 Selective Posterior Calculation

As for the practical issue, a few remarks should be made concerning the computational cost of the EBW algorithm. The computational complexity of calculating posteriors in the EBW algorithm is, for labeled data, $O(N_l)$; while, for unlabeled data, it is $O(N_u \cdot Y)$. In the re-estimation formula for c_{ij}^{y} (2.23) for example, the posterior (2.18) must be calculated for all combinations of classes, states, and Gaussian components (Y, U^y, and K) for each sequence. In the case of the experiment in 2.6.3, $Y = 48$, $U^y = 3$, and $K = 500$; thus, the number of combinations is $72,000$. Since such a large amount of computation is sometimes impractical, we introduce the following approximate calculation, which we call selective posterior calculation.

First, class posteriors (2.20) are calculated for all classes. Next, according to their values, the classes are sorted in descending order (e.g. If $Y=4$, $P(1|X)=0.3, P(2|X)=0.7, P(3|X)=0.1$, and $P(4|X)=0.5$, we have an ordered class index set as $\{2,4,1,3\}$). Posteriors (2.13)–(2.19) are calculated only for the top $M(\ll Y)$ classes and the posterior values for the remaining

classes are set at zero. For instance, if we set $M=3$, there are $4,500$ parameter combinations for (2.18), which is $1/16$ of the original number of combinations.

2.5.5 Classification

Once the ETM-HMM has been trained based on the maximum likelihood principle, unknown sequential data are classified to the class with the largest posterior probability. The class y^* of an unseen sequence X^* is determined by the following formula:

$$y^* = \arg\max_y P(y|\ X^*, \hat{\Theta}),$$

where $\hat{\Theta}$ is the estimate of a set of parameters obtained by the EBW algorithm.

2.6. Experiments

2.6.1 Experimental Conditions

We experimentally validated the proposed algorithm on two datasets: gesture data and speech data. Our goal is to improve the classifiers that learned poorly due to the scarcity of labeled data by adding unlabeled data. The classification error rate (CER) was used to evaluate the performance of the learned classifiers. The data in the original datasets were all labeled; thus, unlabeled data were created by hiding their class labels for experimental purposes. In both experiments, in addition to the few initial labeled training data $\mathcal{D}_l^{\text{ini}}$, either labeled data $\mathcal{D}_l$ or unlabeled data $\mathcal{D}_u$ were added to the training data set. Here, we say that the initial data are "few" when the addition of *labeled* data to the initial training data set decreases the CER on the test data set. This situation implies that the initial labeled training data are insufficient relative to the number of model parameters and the model parameters are not reliably estimated.

Once we found that there were few initial labeled data, such ETM-HMM was trained on the larger quantity of labeled data ($\mathcal{D}_l^{\text{ini}} \cup \mathcal{D}_l$) or on the mixed data ($\mathcal{D}_l^{\text{ini}} \cup \mathcal{D}_u$) by using the EBW algorithm varying the quantity of additional data. The classification performance of learned ETM-HMMs was compared for two types of additional data, labeled or unlabeled, with respect to their quantity. In general, as the quantity of labeled training data increases, the generalization performance improves [51]. Therefore, the addition of labeled data can be regarded as the ideal setting for performance improvement; we can examine how close the performance with the addition of unlabeled data is to the performance with the addition of labeled data.

In addition to the quantity of training data, the classification performance is influenced by two other factors: variances in the initialization of the model parameters and those in the training data selection. Although both kinds of variance should be averaged, since the amount of computation required was too large, we only averaged the effect of data selection over ten trials and used fixed initial model parameters for all trials. For each quantity of additional labeled and unlabeled data, the training data were drawn 10 times randomly from the whole

available training data set. Then, ten different ETM-HMMs were trained on these 10 data subsets. The median of their CERs for the independent test data set was calculated.

Throughout the experiments in this article, we used left-to-right HMMs. Fixed model parameters of those HMMs were set as follows. Class priors, transition probabilities, and mixture coefficients were initialized to uniform distributions. Gaussian means were determined by the k-means algorithm for the whole available training data of all classes. Gaussian covariances were determined by the Voronoi partitions of the data based on the result of the k-means algorithm. The covariance matrices were diagonal. Note that in ETM-HMMs, since the feature spaces are tied, all data can be used for estimating the Gaussian parameters. By undertaking the initialization with large quantities of data, we avoided the effect of poor parameterization so that we could focus our attention on the effect of the data amount.

2.6.2 Gesture Classification

Sign Language Dataset

The first experiment was on gesture data. Each gesture was one of the 15 Japanese sign language (JSL) signs. The signs were "aisatsu (greeting)", "aida (gap)", "au (meet)", "akarui (bright)", "atatakai (warm)", "atarashii (new)", "atarimae (common)", "ataru (hit)", "atsumaru (gather)", "aratamete (anew)", "arigatou (thank you)", "anshin (peace of mind)", "ie (house)", "issho (together)", and "itsumo (always)". Although all of them begin with the same vocal sounds of either 'a' or 'i' when voiced, their hand movements as signs are substantially different. With an electromagnetic position tracking system (Polhemus 3SPACE FASTRAK system) [52] the movements of the hands in three dimensional space and the rotation angles around three axes were measured at a 30 Hz sampling rate. The collected sequences were, therefore, 15 classes ($Y = 15$) and 12 dimensional. Each sign was performed 40 times (30 for training data and 10 for test data) by 20 non-native JSL signers. The total amount of training data was 9000 and the total amount of test data was 3000. All 15 classes have the same amount of data (600 training data and 200 test data for each class). The mean, maximum, and minimum lengths of the sequences were, respectively, 25.6, 44, and 15 for the training data and 24.6, 41, and 16 for the test data.

Preliminary Experiment

Unless the addition of labeled data reduces the CER, unlabeled data cannot reduce the errors, either. Therefore, in our preliminary experiments, we first searched for a situation where the training data were insufficient. Let N_y^{ini} be the amount of initial labeled data for class y. We found that when $N_y^{\mathrm{ini}} = 2$ for all classes, the number of states $U^y = 5$ for all classes, and the number of components $K = 50$, the median of CERs decreased more than 40 points when labeled sequences were added to the training data. This implies that N_y^{ini} is too small relative to the number of model parameters to be estimated. Using the above case as an example, we evaluated the EBW algorithm when unlabeled data were added.

Experimental Results

We trained ETM-HMMs of the structure specified by $U^y = 5$ and $K = 50$. The initial labeled data, $N_y^{\text{ini}} = 2$ for all ys, comprised about 0.33% of the total available training data. Either 150 labeled or unlabeled sequences were added at a time to $\mathcal{D}_l^{\text{ini}}$. That is, the number of additional data, N_l or N_u, was 150. For each number, N_l or N_u, we created ten training data sets by random sampling, and for the ten ETM-HMMs learned from those training data subsets, the CERs on the test data with a size $N_t = 3000$ were computed.

The result of the experiment is shown in Fig. 2.4. Each bar in the graph represents the median of the CERs of ten ETM-HMMs. When no data were added, the median of the CERs was 63.1% as shown by the leftmost bars. Black bars show the change in the CERs caused by the addition of labeled data. The median of CERs decreased to 17.2% at their lowest. White bars show the change in the CERs caused by the addition of unlabeled data. The median of CERs decreased to 50.8% at their lowest. As Gaussian parameters change more than other parameters by adding unlabeled data, we presume that the improved estimation of Gaussian parameters is the most important source of performance improvement.

It should be noted that the addition of labeled data lowered the CERs dramatically, whereas the addition of unlabeled data lowered the CERs gradually. That is, in terms of reducing errors, the labeled data were clearly superior to the unlabeled data. However, we do not usually have additional expensive labeled data and without adding unlabeled data, the median of CERs remains at 63.1%. In this regard, we can say that the addition of unlabeled data by the EBW algorithm was beneficial in improving the classifier for this gesture dataset when the amount of labeled data was limited.

Note that since all classes were equal as regards the number of initial labeled data, the class priors were uniform. Therefore, the initial ETM-HMMs learned by the EBW algorithm were the same as the initial TM-HMMs learned by the BW algorithm.

Comparison with Naive Labeling Approach

We compare our EBW algorithm with the NL approach explained in Section 2.3 in terms of the degree of improvement. Varying the confidence threshold C among $\{0, 0.8, 1.0\}$, we computed the changes in the CERs caused by the NL approach. Here, $C = 0$ indicates the NL approach without a confidence measure. The result is shown in Fig. 2.5 in which the CERs obtained by the EBW algorithm are cited from Fig. 2.4. Although the CERs did not decrease monotonically, the EBW algorithm was able to improve the classification performance in general. In contrast, the change in the CERs caused by the NL approach was unstable. For some C and N_u, the CERs became worse than that of the initial model. From the results, it can be said that the EBW algorithm was superior to the NL approach for the JSL data.

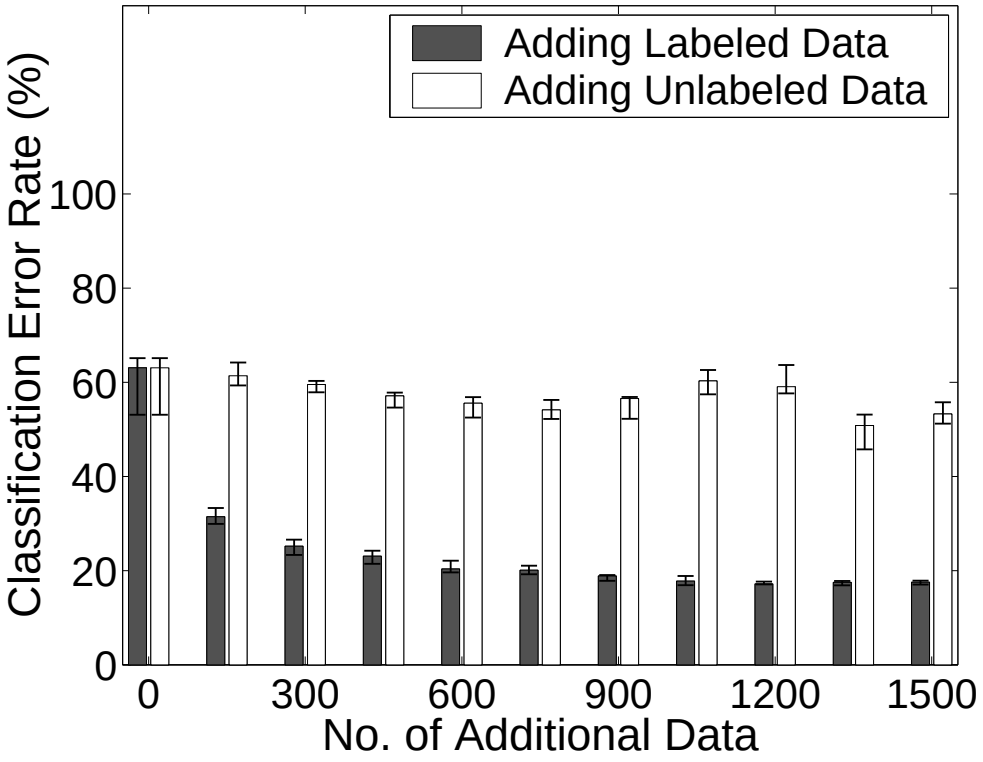


Figure 2.4. The change in the classification error rates for JSL data by the number of additional labeled or unlabeled data. The thick bars represent medians of CERs and the thin lines represent upper and lower quartiles. The initial training data were 2 labeled data for each class (i.e., $N_l^{\mathrm{ini}} = 30$), the number of states $U^y = 5$ for each class, and the number of Gaussians $K = 50$. Either 150 labeled or unlabeled data were added at a time.

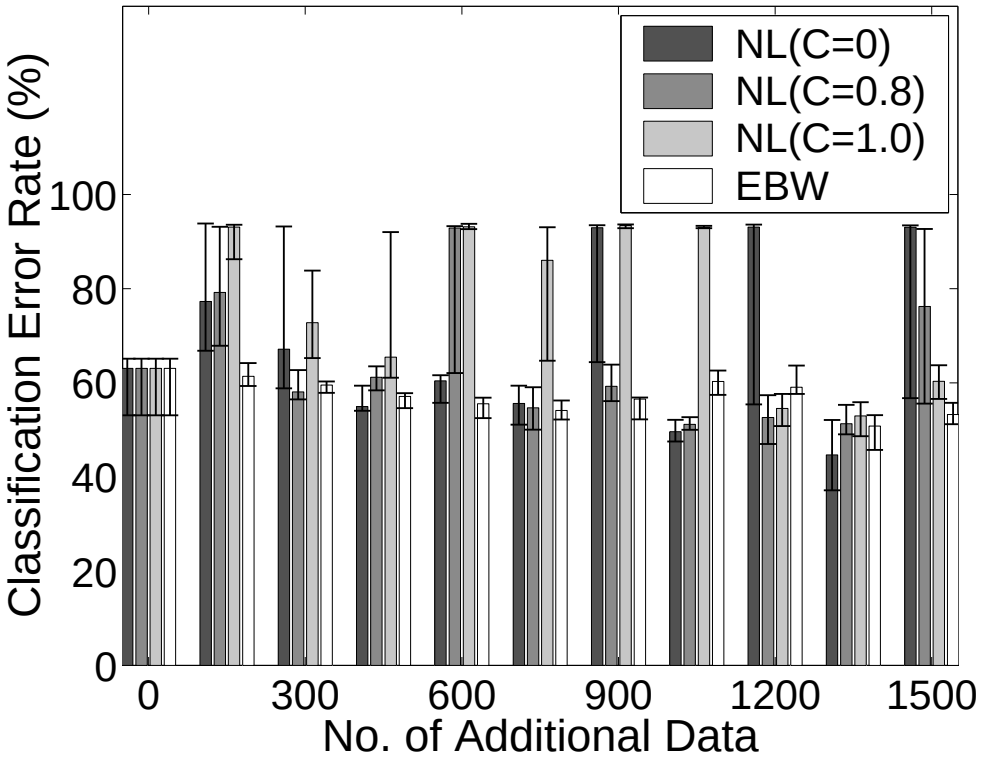


Figure 2.5. The change in the classification error rates for JSL data by the EBW algorithm and by the NL approach for the same amount of unlabeled data. The initial training data were 2 labeled data for each class (i.e., $N_l^{\rm ini}=30$), the number of states $U^y=5$ for each class, and the number of Gaussians $K=50$. Either 150 labeled or unlabeled data were added at a time. The thick bars represent medians of CERs and the thin lines represent upper and lower quartiles. In the NL approach, the threshold of the confidence measure was changed ($C=0, 0.8, 1.0$).

2.6.3 Phoneme Classification

Speech Dataset

As an example of larger data and uneven class distributions, we used the TIMIT corpus [53] of read speech. The phoneme classification tasks on TIMIT have frequently been used to evaluate classifiers for sequential data.

In our experiment, for the training data, we used the standard datasets SX and SI defined in TIMIT. There were 140,099 sequences or phonemes in this training data set, and they were all used for model initialization. For the test data, we used the core test dataset defined in TIMIT. There were 50,754 sequences or phonemes in this test data set. The mean, maximum, and minimum lengths of the sequences were, respectively, 8.9, 238, and 3 for the training data, and 9, 465, and 3 for the test data. In contrast to the gesture data, each class contained different numbers of sequences: from the smallest (149 sequences) to the largest (12,516 sequences) (See Appendix D for detail). As in [54], for training, we grouped the original 64 phoneme categories into 48 as follows: {q → 'remove'}, {ux → uw}, {axr → er}, {ax-h → ah}, {em → m}, {nx → n}, {eng → ng}, {hv → hh}, {pcl, tcl, kcl → cl}, {bcl, dcl, gcl → vcl}, {h#, pau → sil}. As in [54], for testing, we grouped the above 48 phoneme categories into 39 as follows: {cl, vcl, ep → sil}, {el → l}, {en → n}, {zh → sh}, {ao → aa}, {ix → ih}, {ax → ah}. Thirty-nine dimensional feature vectors were extracted as in [55]: 12 MFCC coefficients, log-energy, and the corresponding delta and delta-delta coefficients were computed at a 10 ms frame rate, using a 25 ms Hamming window.

K-means Algorithm with Pruning

As explained in Section 2.6.1, we used the k-means algorithm to initialize Gaussian parameters. In this experiment, since we used a large number of components as shown below, an ordinal k-means algorithm generate many components that contain no data. To avoid such skewed distributions, we used a modified k-means algorithm, which we call the k-means algorithm with pruning. In this variant, components whose numbers of member data are smaller than a pre-defined threshold are eliminated during iteration. This initialization, thus, guarantees that each component has more than a predefined number of data. To initialize mixtures of Gaussian parameters in HMMs, the Linde Buzo Gray (LBG) algorithm [56] is often used. However, the LBG algorithm takes much longer time than our method thus from a practical standpoint, we did not adopt the LBG algorithm. ISODATA [57] has been proposed to overcome this data imbalance among clusters. It adds and deletes components one-by-one, whereas the method we used only deletes (usually more than one at a time) components.

Preliminary Experiment

In the preliminary experiment, as well as the previous experiment on the gesture data, we searched for a situation where there was little training data relative to the number of free model parameters. As a result, we found that the median of CERs decreased more than 20

points when we added labeled data, when $N_y^{\text{ini}} = 5$, $U^y = 3$ for all classes, and $K = 500$. We focused on this case as an example, and evaluated the effect of unlabeled data utilized by the EBW algorithm.

We also examined the effect of M which is introduced in 2.5.4. Varying M among $\{1, 3, 5\}$, we compared the CERs of the learned ETM-HMMs. Since we did not observe any clear improvement by increasing M, we concluded that, as far as the TIMIT corpus is concerned, the choice of M does not affect the performance. Therefore, to minimize the amount of computation, we chose $M = 1$ for the rest of the experiments.

Experimental Results

For $\mathcal{D}_l^{\text{ini}}$, although the class distributions of TIMIT were inhomogeneous, we sampled the data uniformly from all classes. This assumed that we knew there to be 48 classes, but we had no prior knowledge of their distributions. The number of initial labeled data $N_y^{\text{ini}} = 5$ for all y, which comprised about 0.17% of all the available training data. In contrast to the labeled data, the unlabeled data were randomly drawn from the real distribution of the whole training data, since we usually collect unlabeled data without knowing their true classes. The sampled unlabeled data might reflect the true distribution of the labeled data if the amount were large enough. Either 480 labeled or unlabeled data were added at a time until the total reached 4800. For each additional amount, ten different data sets were created as above. Then, ETM-HMMs ($U^y = 3, K = 500$) were trained on these data sets and tested on the same test data set.

The results of these experiments are shown in Fig. 2.6. The median of the CERs of the ETM-HMMs learned only from initial labeled data was 66.7%. For the ETM-HMMs learned from mixed data containing $4,800$ unlabeled data, the median of CERs decreased to 54.1%. Of course the addition of labeled data was more effective (the median of CERs decreased 44.2%); nevertheless, the improvement provided by the unlabeled data with the EBW algorithm may be valuable since labeled data are usually expensive and not easily available. As JSL gesture data, we presume that the improvement comes from the better estimates of Gaussian parameters.

In this experiment, the improvement provided by unlabeled data was more significant than that for the gesture data. One reason may be the difference in the dimensionalities of the two data sets. In [25], it is argued that unlabeled data are more effective when the feature dimensionality is high. The dimensionality of TIMIT data is about three times higher than that of JSL data.

Unfortunately, a significant improvement could not always be achieved by using unlabeled data. When relatively larger initial labeled data were available, the addition of unlabeled data did not reduce the errors. Figure 2.7 shows the case where $N_y^{\text{ini}} = 50$. As with the previous case, $U^y = 3$ and $K = 500$. As can be seen, the addition of unlabeled data did not improve the performance; on the contrary, the addition sometimes had a detrimental effect. For example, when $4,800$ unlabeled data were added, the median of CERs was 44.2% while the initial median of CERs was 42.6%. Here, it should be noted that the addition of labeled data did not improve the performance significantly either, although they did not degrade the performance. The main

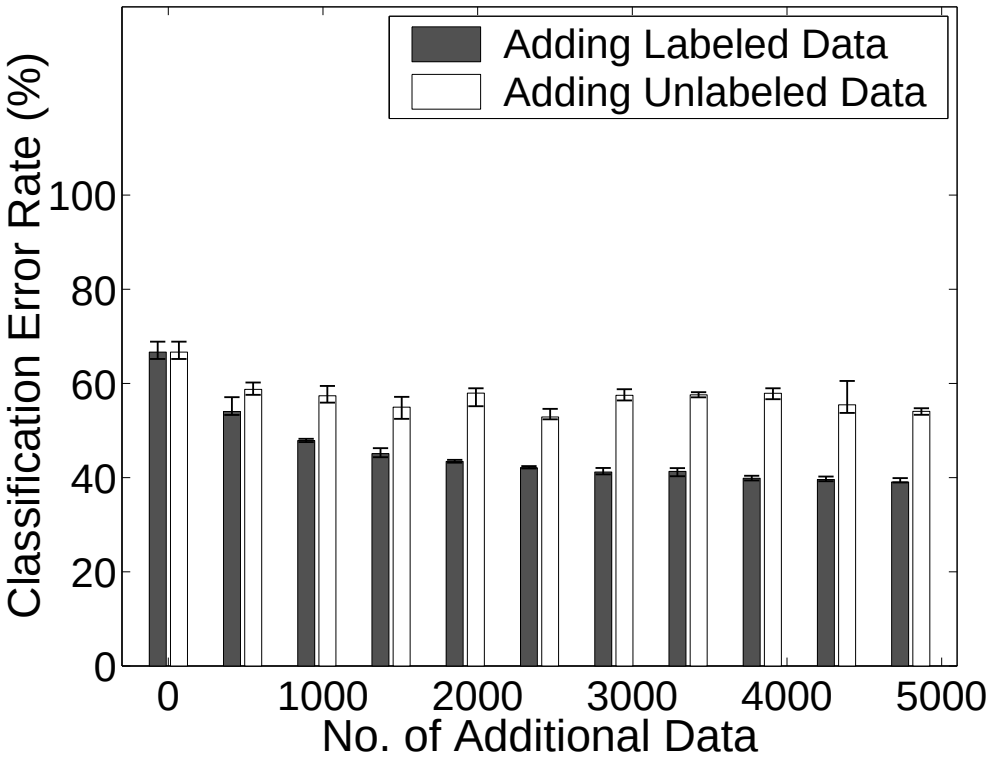


Figure 2.6. The change in the classification error rates for TIMIT data when either 480 labeled or unlabeled sequences were added at a time to the initial training data set ($N_l^{\text{ini}} = 240$). The thick bars represent medians of CERs and the thin lines represent upper and lower quartiles. The number of states $U^y = 3$ for each class and the number of Gaussians $K = 500$.

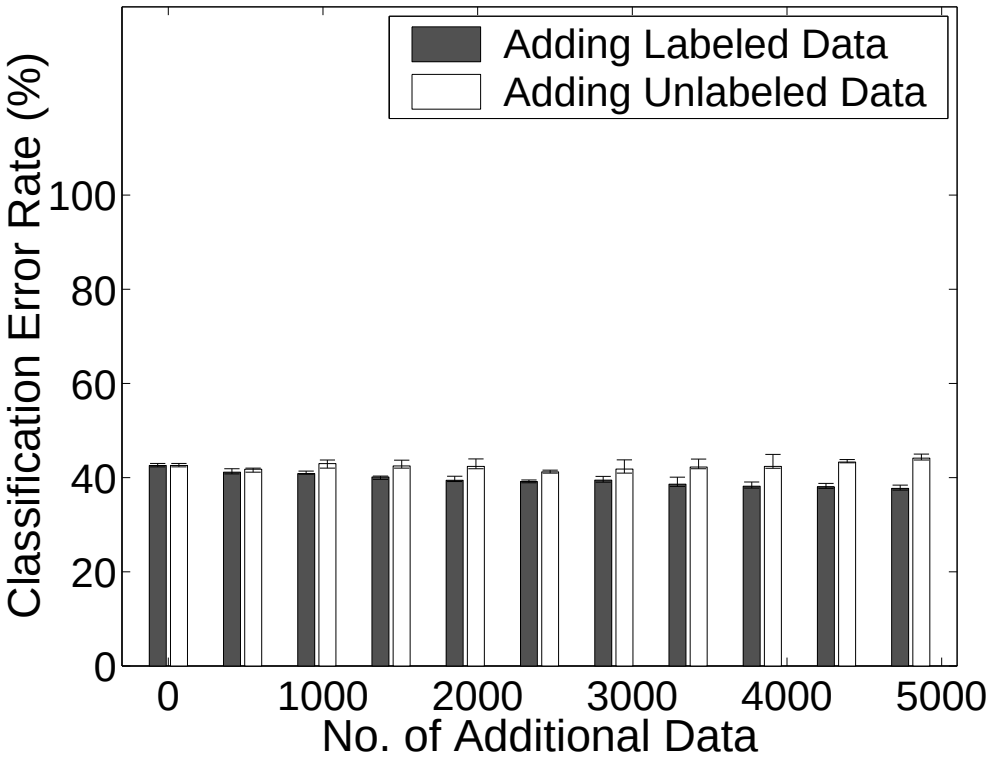


Figure 2.7. The change in the classification error rates for TIMIT data when either 480 labeled or unlabeled sequences were added at a time to the initial training data set ($N_l^{\text{ini}} = 2400$). The thick bars represent medians of CERs and the thin lines represent upper and lower quartiles. The number of states $U^y = 3$ for each class and the number of Gaussians $K = 500$.

reason for the performance degradation may be that the models responsible for the different classes were close to each other as a result of the addition of unlabeled data. That is, the class boundary provided by the sufficient amount of initial labeled data may become blurred through the addition of unlabeled data.

In conclusion, the addition of unlabeled data by the EBW algorithm seems to be useful when HMMs need to be complex to achieve satisfactory performances but labeled data are too scarce to estimate their parameters accurately. In contrast, it may be not helpful when there are enough labeled data available.

Comparison with Naive Labeling Approach

We examined the NL approach using the same phoneme data and model structure ($U^y = 3, K = 500$). Varying C among $\{0, 0.8, 1.0\}$, we computed the changes in the CERs when unlabeled data were added. Figure 2.8 shows the medians of the CERs for 10 data subsets for each amount of added data. For ease of comparison, the results obtained with the EBW algorithm are cited from Fig. 2.6. Clearly, the performance of the ETM-HMMs learned by the EBW algorithm was better than that with the NL approach. This result implies an advantage of our approach. In addition to the higher CERs, the results of the NL approach were unstable: for some C and N_u, the performance degraded from that of the initial ETM-HMMs. This difference in stability suggests another advantage of our approach over the NL approach.

Here, we discuss the possible reasons for the ineffectiveness of the NL approach. In the above experiment, the CER for the initial model was 66.7%. We may regard this CER as indicating poor performance. The pseudo labels generated by such a classifier must be unreliable and the addition of data might have adverse effects. Throughout the experiment, regardless of the amount of additional data, the poor initial parameter estimates were used for the NL approach; in contrast, for the EBW algorithm, both labeled and unlabeled data can be used from the beginning of the learning process. Possibly, the NL approach may work when the initial models are relatively well trained based solely on initial labeled data. When the initial models are unreliable and the purpose of using unlabeled data is to improve the classification to an acceptable level, as has been considered in this paper, our method may work better than the NL approach.

Discussion

The above experiments showed the benefit of unlabeled sequences in designing classifiers. They also suggested notable differences in the utility of labeled and unlabeled sequences. To narrow the gap between the values of these two types of data, we should apply more elaborated usages of unlabeled data. Constraint on class distribution has been used to avoid the degradation caused by unlabeled data [58]. However, in our preliminary experiments, such constraints did not improve the performance on the data set used here. Although we just fixed class prior probabilities rather than calibrate them as in [58], we do not consider that this method could work in our settings where posterior probability dominates class prior probabilities. In

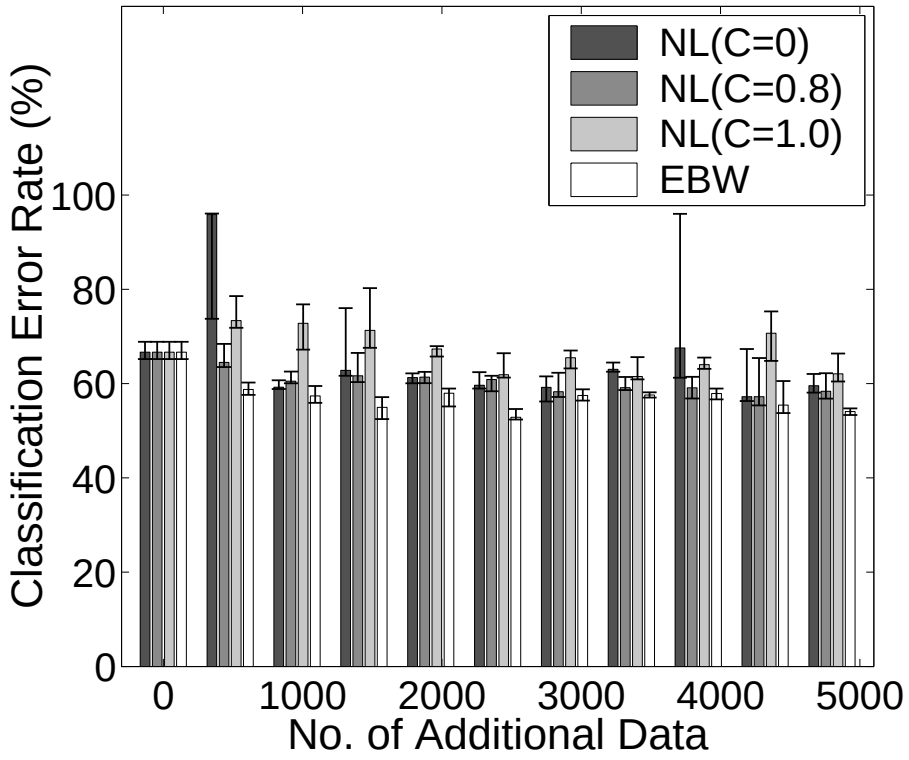


Figure 2.8. The change in the classification error rates for TIMIT data by the EBW algorithm and by the NL approach for the same amount of data. The number of states $U^y = 3$ for each class and the number of Gaussians $K = 500$. Either 480 labeled or unlabeled sequences were added at a time to the initial training data set ($N_l^{\text{ini}} = 240$). The thick bars represent medians of CERs and the thin lines represent upper and lower quartiles. In the NL approach, the confidence measure threshold was changed ($C = 0, 0.8, 1.0$).

contrast, scheduling constraints on the posterior probabilities seems more promising. That is, their values are controlled so that they are not as distinctive between classes at the earlier stage of the learning; instead, they gradually become acute at a later stage. Another important method is the discounting of unlabeled data. As shown in Fig. 2.7, use of unlabeled sequences was not always beneficial. The degradation of classification performance has also been reported for static data [27], [59]. However, by choosing an appropriate weighting factor on unlabeled data so that the contribution of unlabeled data can be reduced, some of such adverse effect could have been avoided. When necessary, a similar discounting technique can be used in the learning of sequence classifiers.

2.7. Summary of This Chapter

In this chapter, we proposed the EBW algorithm to enable the learning of HMMs from both labeled and unlabeled sequential data. Conventionally, in HMM learning, unlabeled sequences have been used heuristically by the NL approach without the guarantee of convergence. In contrast, in the EBW algorithm, the parameter re-estimation formulae have been formally derived in the framework of the EM algorithm. We also found that our method utilized unlabeled sequences more effectively than the NL approach in terms of classification performance. Two experimental results on gesture data and speech data showed that the EBW algorithm reduced the classification errors in most cases in contrast to the NL approach.

Although when the initial labeled data were scarce, our method could compensate for the insufficiency of labeled training data by using unlabeled data, when the initial labeled data were sufficient, the EBW algorithm sometimes had a detrimental effect on the classification performance. This is a limitation of our approach and the situations in which unlabeled sequences do not help should be studied further. The reason why adding unlabeled data could not monotonically improve the performance can probably be explained based on the analysis in [60].

In future work, we can apply our method for adaptation where unseen test data whose properties are different from those of training data can be regarded as unlabeled data [26], [61]. Furthermore, there exists a more sophisticated NL approach called co-training [20], in which feature vectors need to be separated into two feature sets, each of which is capable of learning a classifier. For example, in automatic speech recognition, visual information taken from speaker's mouth region has been used with conventional audio information [62]. When we have such a redundant dataset, it is interesting to compare the NL approach, the EBW algorithm, and the co-training algorithm.

Chapter 3

Image Retrieval by Textual Query

3.1. Introduction to This Chapter

3.1.1 Image Retrieval

In this chaper, we consider the problem of information retrieval (IR). Among the various IR tasks, we are interested in image retrieval. Textual information retrieval has long been the main topic of IR [63]. In contrast, image retrieval is relatively new research field since most data stored previously had been textual data and accumulations of multimedia data including visual data has come together with recent developments of various recording devices. Another reason is that symbols such as texts can be handled more easily than signals by computers. Thus, image retrieval is considered to be difficult.

Currently, in the field of image retrieval, the query by example (QbE) framework has attracted many researchers. In QbE, users are assumed to have an initial query image at hand as an example. The IR system returns the ranked list of images in the order of similarities to the query image[64]. The QbE problem includes the popular research topic known as content-based image retrieval (CBIR) [65]. Roughly speaking, the research on CBIR has resulted in the extraction of good features from images and the exploration of metrics that define how two images are semantically similar despite their different visual appearances. We agree with that QbE can be applied to a variety of areas, but rather than QbE, we studied query by text (QbT) for the following two reasons:

- Textual queries are always available because we can generate them freely. On the other hand, sample images are not always available and we cannot generate them freely.
- The easiest way to acquire sample images to initiate QbE is via QbT.
- Few studies have been conducted on the topic despite its importance.

3.1.2 Cross-Media Information Retrieval

In QbT IR, if documents are signals such as images rather than symbols like word tokens, the task becomes cross-media one: queries and documents are represented by different media. We regard image retrieval by textual queries as typical and at the same time a difficult example of cross-media IR. The typicalities and difficulties are the reasons why we focus on this QbT image retrieval. QbT image retrieval is typical because we often want to use textual queries even for image retrieval as explained in Section 3.1.1. QbT image retrieval is a difficult cross-media IR because the correspondence between queries and target documents is not clear. Some other cross-media IR might, however, be easier; imagine another example of cross-media IR where queries are musical scores and documents are musical sounds. Their correspondence may be more apparent than that of QbT image retrieval.

Various approaches are possible for QbT image retrieval. The most straightforward one is to learn the relationships between textual symbols and image signals. However, to achieve this we need a bunch of training data collected and enormously complicated model structures to learn. This is almost equivalent to building artificial intelligence with human common sense. We leave this high-flying attempt for the future, and restrict our attention to a less ambitious setting.

3.1.3 Annotation-Based Image Retrieval

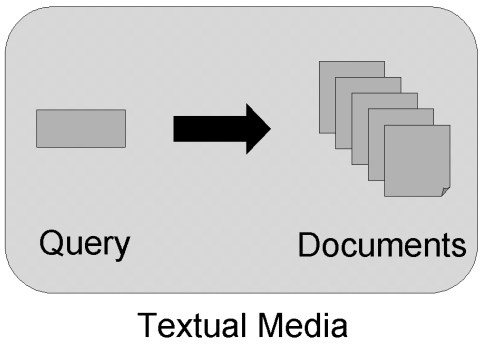
In this thesis, we assume images are annotated by some keywords (annotations) that explain the contents of images. This condition means that images have already been translated to a textual medium. Since the medium of annotations is text and that of queries is also text, QbT image retrieval can now be seen as textual IR (See Fig. 3.1). Such relaxation of cross-media IR seems to simplify the task, but, in fact, the task is still difficult. The details of this difficulty are explained in Section 3.3.1.

In spite of its practical advantages, there have been few research works conducted on annotation-based cross-media IR. Exceptional examples are found in captioned image retrieval where natural language processing (NLP) techniques were employed [34], [66]. Another example is retrieving audio data using an ontology, which is an explicit specification of the relationships between words [67]. Our work is different from these previous researches in that we use multiple information sources to improve the retrieval model.

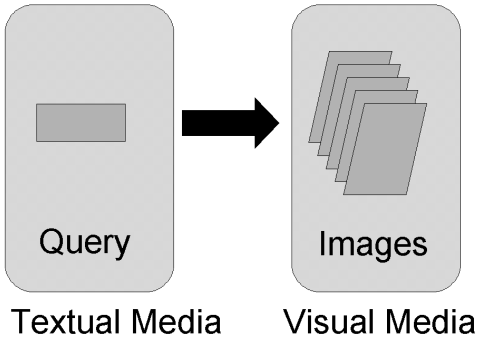
3.2. Basic Information Retrieval Model

3.2.1 Language Model-based Information Retrieval

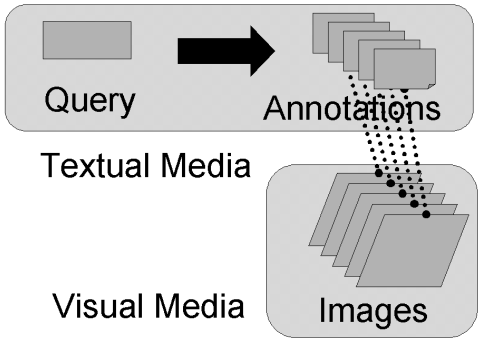
Although the traditional vector space model (VSM) [63] is the most widely known framework in IR, recent researches have based most text IR models on language models (LMs) [68] because the LM framework is theoretically rigorous and effective in practice.



(a) Textual Information Retrieval



(b) Cross-Media Information Retrieval



(c) Annotation-Based Cross-Media Information Retrieval

Figure 3.1. This figure shows schematic diagrams of different types of information retrieval. a) In conventional textual IR, queries are texts (keywords) and documents are texts, too. b) In cross-media visual IR, queries are texts (keywords) and documents are images. c) In annotation-based cross-media visual IR, queries are texts (keywords) and documents are both images and their annotations.

Language models are in fact probability distributions over vocabularies —in other words, probabilities for the occurrence of a certain word. The basic LM-based IR system is built from the probabilities of generating words in the vocabulary $\mathcal{V}$ of size V that is defined over all documents. The simplest formulation of LMs is based on unigram models where the context of term appearance does not count:

$$P(t_1, t_2, t_3, ...) = \prod_i P(t_i), \tag{3.1}$$

where t_i represents the ith term in the observation.

Obviously, imposing dependencies between terms using a tool such as the bigram model

$$P(t_1, t_2, t_3, ...) = P(t_1) \prod_i P(t_i|t_{i-1}) \tag{3.2}$$

leads to richer expressions but requires more information to learn the model accurately. Throughout this thesis, we use unigrams because what we are considering is not ordinary text but annotations, which are likely formulated as a list of keywords. That is, the order of terms does not have any particular meaning in annotation-based image retrieval.

Let D be the set of all N documents called a collection. For each query, documents are ranked according to their relevance to the query. Let q_i be a query term and d_j be annotation term. In LM-based IR, relevances are determined as the likelihoods of query $\mathbf{q} =< q_1, q_2, ..., q_i, ..., q_L >$ of length L given a nth document $\mathbf{d}_n =< d_1, d_2, ..., d_j, ..., d_M >$ of length M:

$$P(\mathbf{q}|\mathbf{d}_n) \quad = \quad \prod_i P(q_i|\mathbf{d}_n). \tag{3.3}$$

Because an LM is defined on one document rather than across the whole document collection, we call each probability distribution defined over a document a document model (DM):

$$DM_n \quad = \quad < P(w_1|\mathbf{d}_n), P(w_2|\mathbf{d}_n), ..., P(w_V|\mathbf{d}_n) > . \tag{3.4}$$

The parameters of the nth DM are estimated by the maximum likelihood (ML) estimation as follows:

$$P(w_v|\mathbf{d}_n) \quad = \quad \frac{n(w_v) \in \mathbf{d}_n}{\sum_{n'} n(w_v) \in \mathbf{d}_{n'}}, \tag{3.5}$$

where $n(w_v)$ represents the frequency of the word w_v.

In many cases of textual retrieval, an additional term $P(w|D)$, called the collection model (CM), is combined with (3.3) heuristically, and this addition has indicated performance improvement [69]. Let λ be a weighting parameter. The CM is used as follows:

$$P(\mathbf{q}|\mathbf{d}_n) \quad = \quad \prod_i \{\lambda P(q_i|\mathbf{d}_n) + (1 - \lambda) P(q_i|D)\} . \tag{3.6}$$

In this thesis, however, we do not use this smoothing term since the validity of this heuristics in cross-media IR has not been clarified. Moreover, we want to focus our attention on the most basic model at this time.

3.3. Exploiting Word Association

3.3.1 Vocabulary Problem

IR systems have to deal with ambiguities inherent in natural languages. The distinction made by Rabitti and Savino between the conventional "database approach" and "IR approach" is important [70]. The database approach deals with queries that precisely define the values of all attributes in data. The task is to evaluate the Boolean combination of attributes to find a "true" document. In contrast, the IR approach should rank documents according to the similarities among unstructured documents with the unrestricted query. Actually, the biggest problem in IR is considered to be the lexical mismatch between the query words and annotation words; even if both query and annotation words refer to the same concept, there are many expressions possible for that single concept. Such a problem is called a vocabulary problem [71]. This problem is derived from the variety in our word usage; more specifically, synonymy. In most IRs, including textual IR and annotation-based cross-media IR, it is clear that simple term-matching strategies (the database approach) will not work because of this problem. Some solutions have been proposed in the textual IR field, and the most promising solution so far is to acquire word associations from data. By using the learned relationships, queries and texts in the target documents can be semantically connected.

3.3.2 Query Expansion

One solution for the vocabulary problem is use of a technique called query expansion (QE) [72], [73]. In QE, a query is expanded by adding the synonyms to the initial query. QE is implemented with the technique known as relevance feedback (RF) [74].

Originally, RF was introduced to change the weights of words, determining which word in the vocabulary is important and which one is not, according to the users' responses. Users are expected to evaluate the initial ranked list of retrieved documents by specifying whether each of these documents is relevant or irrelevant. By expending the additional effort to use RF, a user can refine the importance weights of words to be close to their information needs. Later, RF is used for the QE. In RF-based QE, the words contained in the relevant documents, whose relevances are determined by users, are added to the original query.

Although RF is effective, some user studies suggest that casual users are not willing to use RF due to its cumbersome additional operations. For this reason, RF is now replaced by pseudo-RF (PRF) [75] and other methods that perform RF-like operations automatically without bothering users. Despite its recent unpopularity in text retrieval, RF is still a preferred technique in the field of image retrieval, where no sound IR architecture has yet been established and interactive refinement is the most reliable procedure. Some machine-learning techniques have been applied to improve the RF in QbE image retrieval. Although RF is also used in the context of learning from multiple information sources [76], [77], and RF is a matter worthy of study, we do not consider RF any further in this article. The reason is that RF induces

cognitive burden on the user and the automation of IR process should be our research goal.

QE with PRF is called automatic QE. In automatic QE, the words contained in the top-ranked instead of relevant documents are concatenated to the original query. The modified query is expected to have words that can be matched with texts in relevant documents. However, it is clear that the performance of initial retrieval plays a crucial role in PRF. Therefore, we consider it more desirable to expand queries automatically by using predefined knowledge without using the retrieval results. For the QE prior to any document ranking, we can use thesauri that define the semantic associations between words. Such thesauri can be introduced externally in the form of knowledge or can be learned from data.

General Thesaurus

The external knowledge considered here are hand-crafted thesauri. A well known example of a general thesaurus is WordNet [78], which is build based on psycholinguistic interest and is not intended for information processing systems. However, since WordNet is designed to be used with computers, it has been used many information processing applications, including IR. Although WordNet sometimes does not work well in textual IR, it has been used successfully in cross-media IR where images are retrieved based on their captions (annotations) [33], [34]. The conclusion in these references is that WordNet is suited for the short texts such as annotations.

As successful as the use of WordNet in cross-media IR was, there are two recognized drawbacks of general thesauri in IR tasks [79]: First, the integration process of external knowledge into the task domain is cumbersome. A data-driven thesaurus is more desirable because it can be built automatically and can be integrated into the probabilistic models directly. Second, they are not specific enough for individual document collection. One particularly serious example is that they lack information about proper nouns. Because domain-specific rather than general thesauri are preferred for IR, our aim here is to realize a learned thesaurus based solely on available data without any manual effort.

Cooccurrence-based Method

If the relationships between words can be learned from the data, queries can be converted into other expressions that can match the texts of target documents. For this purpose, the most heavily studied information source is word cooccurrence in the documents. Cooccurrences are defined as the occcurrences of two words in a single document. The assumption is that two words appearing in the same document have similar meaning. This may sound naive but it works well in practice.

Semantic relationships between words can be represented as a probabilistic model learned from the frequencies of word cooccurrence. The semantic similarity between two words (w_k and w_l in the vocabulary $\mathcal{V}$) is estimated as follows:

$$P(w_l|w_k) = \frac{n(w_k, w_l)}{n(w_k)}, \tag{3.7}$$

where $n(w_k, w_l)$ represents the frequency of cooccurrence w_k and w_l, and $n(w_k)$ represents the frequency of w_k in all documents D. Probabilistic relationships between all word pairs can be calculated in this way. Note that the relationship between two words is asymmetric, and this asymmetry is natural since words have different degrees of abstraction. For example, the concrete word "lion" must be strongly related to the word "animal," whereas the abstract word "animal" is not necessarily strongly related to "lion" since "animal" has a wider variety of relationships with other specific animals and the degree of semantic similarities may diffuse.

3.3.3 Statistical Translation Model

Once we have the word-word relationships, we can include the knowledge in the retrieval model. One successful method of text retrieval derives from statistical translation [80] in which query terms are assumed to be translated into document terms (annotation terms, in our case):

$$\begin{aligned} P(\mathbf{q}|\mathbf{d}) &= \prod_{i \in \mathbf{q}} \sum_{j} P(q_i|d_l, \mathbf{d}_n) P(d_j|\mathbf{d}_n) \\ &\approx \prod_{i \in \mathbf{q}} \sum_{j} P(q_i|d_j) P(d_j|\mathbf{d}_n). \end{aligned} \tag{3.8}$$

As well as the standard LM-based IR, the likelihoods of an nth document generating $\mathbf{q}$ are assumed to be its relevance to the query. We call this model the statistical translation model (STM). The $P(d_j|\mathbf{d}_n)$ in (3.8) is DM as defined in (3.4), and the component that corresponds to $P(q_i|d_j)$ in (3.8) is called the translation model (TM) in [80]. In the reference, the TM takes word orders into consideration. In our model, as explained in Section 3.2.1, the context of word appearance does not matter. In this regard, we call $P(q_i|d_j)$ the word association matrix (WAM) instead of the TM. The WAM is estimated from the word cooccurrence information in annotations using (3.7).

By the approximation made in (3.8), we assume that WAMs are common to all documents, whereas DMs are, of course, document dependent. In our view, an STM performs probabilistic QE using the WAM prior to the retrieval. Note that conventional PRF is not practical for annotation-based image retrieval because annotations are very sparse. As discussed in Section 3.3.2, good initial retrieval is needed to perform PRF. In annotation-based image retrieval, if the WAM is not used, we usually cannot have any matched or pseudo-relevant documents. Thus, the use of top-ranked documents as the relevant documents does not make sense. That is, expansion prior to the retrieval is essential.

3.3.4 Probabilistic Latent Semantic Indexing

There is another approach to dealing with the vocabulary problem, and it operates by indexing each word by latent semantic categories. The categories are estimated so that all synonyms are indexed by the same categories. In other words, all words are grouped based on the latent semantic similarities. One well-known example of this approach is latent semantic indexing (LSI) [5]. As a probabilistic model of LSI, probabilistic Latent Semantic Indexing (pLSI) has been

proposed [81]. Compared to LSI, pLSI enjoys some advantages of probabilistic representation such as robustness to noises. Let z_k be the kth latent index. In pLSI, a query is assumed to be generated according to the following probabilistic model:

$$\begin{aligned} P(\mathbf{q}|\mathbf{d}) &= \prod_{i \in \mathbf{q}} \sum_k P(q_i|z_k, \mathbf{d}_n) P(z_k|\mathbf{d}_n) \\ &\approx \prod_{i \in \mathbf{q}} \sum_k P(q_i|z_k) P(z_k|\mathbf{d}_n). \end{aligned} \tag{3.9}$$

The documents are ranked by the likelihood of generating the query from this model. Since STM and pLSI have similar model hierarchy or complexity, it seems fair to compare these two approaches. Note that (3.9) looks similar to (3.8), but the intervening variables d and z are either evident word tokens or latent semantic categories. Word tokens can be observed directly and the WAM and DM in STM can be learned separately. The latent index must be estimated from data and the entire model parameters are learned simultaneously. For the learning of pLSI, Hofmann proposed the tempered EM (TEM) algorithm to avoid over-fitting; however, in our experiment, we found that TEM based on perplexity measures does not perform well. This result may be explained by the observation made by Azzopardi et al. that the perplexity of LM is a good predictive performance measure but not a good IR performance measure [82]. For this reason, we used a manually tuned early-stopping criterion; that is, we terminated the EM algorithm at the pre-defined appropriate iteration step.

3.4. Use of Image Information

3.4.1 Annotation Insufficiency

Annotations are now thought to be highly useful even if additional human efforts are needed to prepare them. However, providing annotations to images by hand is tedious work and it is less likely that well-annotated images are available in abundance. Therefore, we may expect that images are annotated with only small number of keywords. The worst case could be a single word attached to an image and that word is extracted from the file name of the image. Insufficiency of annotations is surely the major obstacle to performing annotation-based image retrieval.

If the target documents are regular texts, the cooccurrence information may be sufficient to estimate a WAM. However, as explained above, annotation data are very sparse; hence, the cooccurrence information is even more sparse. Consequently, a WAM estimated based solely on annotations may lack a huge portion of information on the relationships between words. We propose a new method to utilize image data that are paired with annotations for the learning of a WAM.

3.4.2 Interpolation by Image Similarities

Although it is virtually impossible to learn the connection between natural language words and image features, we think it is easier to believe that the similarities between images have something to do with the similarities between their annotation words. Image features are usually dense—every dimension of feature vector contains some values —and we can almost always define similarities between images. This motivates us to use these image similarities to help estimate word similarities.

To measure similarity, we use Kullback-Leibler divergence (KLD) [51]. Assuming that the underlying probability distributions generate images and images can be represented by feature histograms, the similarity between two histograms can be calculated as the distance between two distributions. Let $P(\mathbf{i}_i)$ and $P(\mathbf{i}_j)$ be such distributions corresponding to two histograms $\mathbf{i}_i$ and $\mathbf{i}_j$. Then, the KLD between the ith and jth images is:

$$KL\left(P(\mathbf{i}_i)||P(\mathbf{i}_j)\right) = \sum_{\mathbf{i}} P(\mathbf{i}_i) \log \frac{P(\mathbf{i}_i)}{P(\mathbf{i}_j)}. \tag{3.10}$$

KLD is non-negative and zero if and only if $P(\mathbf{i}_i) = P(\mathbf{i}_j)$. If $\mathbf{i}_i = \mathbf{i}_j$, we skip the calculation. This happens when $i = j$ and also when two images are not exactly the same but approximated to be the same during the quantization process of making histograms.

Let $\mathbf{w}_i$ be the fixed length annotation vector for the ith document, $\mathbf{w}_j$ the annotation vector for the jth document:

$$\begin{aligned} \mathbf{w}_i &= < n(w_{i,1}), n(w_{i,2}), ..., n(w_{i,k}), ..., n(w_{i,V}) > (0 \leq k \leq V) \\ \mathbf{w}_j &= < n(w_{j,1}), n(w_{j,2}), ..., n(w_{j,l}), ..., n(w_{j,V}) > (0 \leq l \leq V). \end{aligned}$$

For example, if the second word appears once in the ith annotation, $\mathbf{w}_i =< 0, 1, ..., 0, ..., 0 >$ and if the first and the last words appear once and the second word appears twice in the jth annotation, $\mathbf{w}_j =< 1, 2, ..., 0, ..., 1 >$. Based on the similarity between two images, the similarity $s_{k,l}$ between kth and lth words, each of which appears in ith and jth documents respectively, is calculated by using KLD as follows:

$$s_{k,l}(i,j) = \begin{cases} \frac{1}{KL(P(\mathbf{i}_i)||P(\mathbf{i}_j))} & \text{if } n(w_{i,k}) \geq 1 \wedge n(w_{j,l}) \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

$$s_{kl} = \sum_{ij} s_{k,l}(i,j). \tag{3.12}$$

Note that we do not use the information on word frequencies here, but only consider whether the word exists. This is the simplification we made by utilizing the fact that any given words seldom appear more than once in an annotation.

By calculating all similarity scores between all word pairs in the vocabulary, we have the

similarity matrix:

$$SM = \begin{pmatrix} s_{11} & s_{12} & \dots & s_{1V} \\ s_{21} & s_{22} & \dots & s_{2V} \\ \vdots & \vdots & \ddots & \vdots \\ s_{V1} & s_{V2} & \dots & s_{VV} \end{pmatrix}. \tag{3.13}$$

By normalizing the SM, the probabilities of associations between kth and lth words is obtained:

$$\begin{aligned} P(w_k|w_l) &= \frac{s_{kl}}{\sum_l s_{kl}} \\ &= a_{kl}. \end{aligned} \tag{3.14}$$

Finally, we have the following WAM:

$$WAM = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1V} \\ a_{21} & a_{22} & \dots & a_{2V} \\ \vdots & \vdots & \ddots & \vdots \\ a_{V1} & a_{V2} & \dots & a_{VV} \end{pmatrix}. \tag{3.15}$$

Note that an STM includes a basic DM (3.3) as a special case: if the WAM is an identity matrix, no expansion takes place and an STM will be reduced to a DM.

3.4.3 Feature Expansion by Concatenation

For the latent variable model, the paired information is utilized by combining textual and visual features. The basic idea is quite simple: if we have textual annotation $\mathbf{w}$ and image feature vector $\mathbf{i}$, then the concatenated feature vector $\mathbf{x} =< \mathbf{w}, \mathbf{i} >$. Once the pre-concatenated vectors are replaced with this new feature vectors, the remaining learning process is the same.

Such a method has already been used in [83], in which textual features and visual features are concatenated and LSI is applied for the clustering. In this thesis, we conduct probabilistic versions of that work; that is, we use pLSI (3.9) instead of LSI because pLSI is supposed to work better than LSI. Another example that concatenates two features uses unigram models for both textual and visual information in [84]. Their IR scheme is, however, not QbT in the sense used in this thesis; thus, we cannot make a direct comparison.

3.5. Evaluation of Cross-Media Information Retrieval

3.5.1 Test Collection

Evaluation of image retrieval methods is one of the biggest problems in this field because there is no established *test collection*. A test collection is a set of queries and documents in which relevant documents are manually specified for each query. For textual IR, TREC [1] is one of the

[1] http://trec.nist.gov

most widely used test collections. For QbE image retrieval, efforts are being made to construct a publicly available test collection such as Benchathlon [85][2]. Furthermore, there is an attempt to evaluate the difficulty of QbE tasks based on the complexity of the database itself without preparing test queries [86]. For QbT image retrieval, which we consider here, there seems to be no well-prepared test collection as far as we know except for some inhouse datasets, nor does there exist an evaluation method that does not use test collections. Therefore, we shall prepare a test collection by ourselves or invent an evaluation method that does not require such collection. Building a test collection itself is of course an important research topic, but at this time, we have decided to evaluate the performance of our algorithm by using an easily built synthetic test collection.

3.5.2 Synthetic Test Collection

Since our research goal is to associate query terms with annotations, we can restrict our evaluation to the situation where query terms and annotation terms do not match directly. In other words, the query and the annotation do not contain the same word. In this regard, we take some annotation words randomly from the original annotations and use them as pseudo-queries. For example, if two terms w_1, w_2 are selected from the ith annotation $\mathbf{w}_i =< n(w_{i,1}), n(w_{i,2}), ..., n(w_{i,k}), ..., n(w_{i,V}) >$, the synthetic query $\mathbf{q}_i =< n(q_{i,1}), n(q_{i,2}) >$ and the new annotation $\mathbf{w}_i^{'} =< n(w_{i,1}) - 1, n(w_{i,2}) - 1, ..., n(w_{i,k}), ..., n(w_{i,V}) >$ are created. Since we know from which document such queries are created, we can identify one image that should be relevant to the query. In general, one query has more than one relevant document, and this is indeed true for our dataset. However, we assume there is only one relevant document per query, since we do not have any knowledge about the relationships between queries and other relevant documents.

3.5.3 Performance Measure

We use the synthetic dataset constructed from the original data according to the procedure explained in Section 3.5.2. In practical image retrieval systems, retrieved images may be listed in a computer window. If the window can display ten images at a time, then for example, the top ten images can be viewed in the first page and lower-ranked images are only accessible by turning pages over. Therefore, it is preferable that the relevant image is included on the first page. Here, we use the first-page hit rate (FPHR) as a performance measure. If one page can display 10 or 20 images, we use the notations FPHR10 or FPHR20. Another criterion we used for the evaluation was the averaged ranking of relevant documents over all queries. We sued the median for the average, since we want to avoid the influences of extreme values. We believe the behavior of the median is preferable to the mean in understanding the practicality of IR systems.

[2]http://www.benchathlon.net/

Table 3.1. Summary of three categories in the photo object image dataset.

Category	No. of Documents	No. of Vocabularies
Education	207	260
Sport & Leisure	2303	1165
House & Home	4996	2101

3.6. Experiments

3.6.1 Object Image Data Set

The dataset we used is taken from Hemera Photo Object CDs[3]. Images are color pictures consisting of 14 categories. The photographic subjects are various objects, which are annotated manually. The mean number of annotations is 8.87, maximum number is 25, and minimum number is 1. The annotations mostly refer to the contents of images rather than visual traits.

3.6.2 Image Feature

We used a color histogram for image features. A color histogram represents the proportion of pixels of each color within an image. Color information is sometimes considered a low-level feature since it does not correspond to conceptual contents of images. In our model, we already have semantic information from annotations. Therefore, low-level features may supplement the information sparseness well. As for color space, we used RGB (red, green, blue) space. We divided each dimension of the color space into four subspaces; thus, the dimensionality of the color histogram is 64. In our preliminary experiments, the difference of dimensionality yielded some variation in performance. However, because the degree of those changes were moderate, we did not investigate the influence of dimensionalities to any greater depth.

3.6.3 Experimental Conditions

Among the 14 categories, we choose three categories whose sizes are either largest, medium, smallest. Topics of these categories are "Education," "Sport & Leisure," and "House & Home." These three categories are summarized in Table 3.1. We conducted IR tasks within each category, and to observe the effects of other factors clearly, we pegged the length of query L to two. This number is assumed to be the query length most often used by casual users. A study of query length via the WWW search engine reported that the average query length is 2.35 [87]. This number may be applicable in various IR settings including cross-media IR. We believe casual users of IR systems are not willing to use longer queries.

[3]http://www.hemera.com

3.6.4 Experimental Results

This subsection compares the performances of STM under different conditions. We first compared the three following three model types:

- No word association model (LM), which is capable of term matching only.
- STM whose WAM is estimated without visual information.
- STM whose WAM is estimated with visual information.

All documents have the same number of annotation words in each experimental condition, and we varied the size of annotations M from 1 to 3. Note that when $M=1$, there is no cooccurrence, and the results for LM and that for STM without visual information are the same.

Tables 3.2, 3.3, and 3.4 show the results of IR for three categories. In these tables, since the median values change according to the number of documents in each category, for the category-wide comparison, we list the position of the median value in the collection. If documents are selected randomly regardless of queries, the position of a relevant document will be in the middle (50%) of the collection on average. The use of the WAM improved the performance significantly in the small collection, as shown in Table 3.2. In Tables 3.2 and 3.3, when the number of annotations was small, visual information certainly helped to improve the estimation of WAMs. However, as the size of annotations increased, the relative contributions given by visual information decreased. This is natural because annotations usually contain semantically more meaningful information than visual information in IR.

Although the use of visual information according to our approach was effective for the small collection, when the collection size increased, the addition of visual information did not work well, as shown in Table 3.4. Considering the fact that the position of relevant documents in the large collection was below 10%, the ineffectiveness of visual information may suggest that the sparseness of textual information can be compensated for by using abundant documents. That is, our approach might not leave room for improvement when the performance reaches a certain level. However, from a practical point of view, the FPHRs achieved were still too low and more work is needed to achieve satisfactory retrieval performance from larger image databases.

3.6.5 Comparison with Conventional Method

This subsection compares the performances of STMs and pLSI models. The feature vectors in the pLSI model were concatenated according to Section 3.4.3. That is, each entry in an image histogram was regarded as a word. Among the three categories used, we showed the result for the smallest category, "Education," where the use of visual features had been shown to be helpful in the previous experiments.

As shown in Table 3.5, pLSI models perform worse than our STM-based method, although they could achieve improvements over simple LMs. This difference in performance might be cause by the performance of basic retrieval models. That is, pLSI models are less powerful

Table 3.2. This table shows the retrieval performances of LM, STM without visual information, and STM with visual information when increasing the size of annotations M from 1 to 3. The performances are measured by FPHR10, FPHR20, and the median in the category "Education."

$M = 1$

Model	FPHR10 (%)	FPHR20 (%)	Median (Position (%))
LM: Annotation Only	4.35	6.76	109 (52.66)
STM: Annotation Only	4.35	6.76	109 (52.66)
STM: Annotation and Image	39.13	53.62	17 (8.21)

$M = 2$

Model	FPHR10 (%)	FPHR20 (%)	Median (Position (%))
LM: Annotation Only	3.38	8.21	103 (49.76)
STM: Annotation Only	39.61	54.59	18 (8.70)
STM: Annotation and Image	51.21	69.57	10 (4.83)

$M = 3$

Model	FPHR10 (%)	FPHR20 (%)	Median (Position (%))
LM: Annotation Only	8.21	17.39	97 (46.86)
STM: Annotation Only	55.56	69.57	9 (4.35)
STM: Annotation and Image	54.11	71.98	10 (4.83)

Table 3.3. This table shows the retrieval performances of LM, STM without visual information, and STM with visual information when increasing the size of annotations M from 1 to 3. The performances are measured by FPHR10, FPHR20, and the median in the category "Sport & Leisure."

$M = 1$

Model	FPHR10 (%)	FPHR20 (%)	Median (Position (%))
LM: Annotation Only	0.83	1.09	1156.5 (50.24)
STM: Annotation Only	0.83	1.09	1156.5 (50.24)
STM: Annotation and Image	10.08	15.33	212 (9.21)

$M = 2$

Model	FPHR10 (%)	FPHR20 (%)	Median (Position (%))
LM: Annotation Only	0.65	1.22	1136 (49.35)
STM: Annotation Only	13.16	20.07	137 (5.95)
STM: Annotation and Image	15.64	23.68	95 (4.13)

$M = 3$

Model	FPHR10 (%)	FPHR20 (%)	Median (Position (%))
LM: Annotation Only	1.87	2.87	1133 (49.22)
STM: Annotation Only	23.20	34.58	44 (1.91)
STM: Annotation and Image	16.20	25.46	70 (3.04)

Table 3.4. This table shows the retrieval performances of LM, STM without visual information, and STM with visual information when increasing the size of annotations M from 1 to 3. The performances are measured by FPHR10, FPHR20, and the median in the category "House & Home."

$M = 1$

Model	FPHR10 (%)	FPHR20 (%)	Median (Position (%))
LM: Annotation Only	0.30	0.48	2513 (50.30)
STM: Annotation Only	0.30	0.48	2513 (50.30)
STM: Annotation and Image	0.70	1.18	1587 (31.77)

$M = 2$

Model	FPHR10 (%)	FPHR20 (%)	Median (Position (%))
LM: Annotation Only	0.68	0.98	2486.5 (49.77)
STM: Annotation Only	8.61	13.61	312 (6.25)
STM: Annotation and Image	6.08	9.07	457 (9.15)

$M = 3$

Model	FPHR10 (%)	FPHR20 (%)	Median (Position (%))
LM: Annotation Only	1.66	2.28	2473.5 (49.51)
STM: Annotation Only	16.57	24.64	100 (2.00)
STM: Annotation and Image	7.59	11.27	325.5 (6.52)

than STMs as far as this experiment concerns. Another reason for this result could be derived from the limitation in using textual and image features, assuming their independence. In our method, visual information was used to interpolate the sparseness of textual information. The main information source was annotations and visual information was not equally used but used as subsidiary information to learn the model.

3.7. Summary of This Chapter

In this chapter, we have described the difficulty of QbE cross-media image retrieval even if annotations are provided. We clarified that in order to retrieve images with annotations, some forms of word associations are needed. In an effort to solve this problem, we introduced an STM that has been used in textual IR, an IR model with such association. The sparseness of annotation, however, prevents us from learning STM directly. To mitigate this problem, we have proposed the use of visual information that is paired with textual information in learning the model. We also have proposed an evaluation method by using synthetic test collection for the annotation-based cross-media IR. Experimental results show improvements in IR performances given by paired data via our method. We also compared our method with another multi-information source method in the context of QbE cross-media IR. Experimental results suggest that our method could use the paired data more effectively. For a larger dataset, however, the behavior of our method was unsatisfactory.

Further research issues for the improvement include the term-weighting scheme to avoid estimating incorrect word associations. Textual IR has been struggling with this term-weighting issue [88] and similar techniques may apply to our model. Another scheme is the combination of learned parameters or matrices. Here, we point out the difference between our method and other method such as the image registration techniques in combining parameters. Image registration is the application area where unsupervised techniques are used as well as IR. Different matrices that represent different images are aligned so as to use all images jointly [89]. However, in IR, in contrast to the registration, there is no particular template on which the rest of the matrices are aligned; some parts of the WAMs learned from textual information are good, and some parts of them from visual information are also good, but we do not know which parts of WAMs are effective in IR. Therefore, in our framework, simple, weighted combinations of two types of WAMs may not work. Actually, preliminary experiments using combined WAMs only exhibited slight improvements or even deterioration, depending on the experimental setting, even if weighting parameters were tuned. Some type of supervision, however, can be used as an information source to extract effective sub-components from two types of WAMs.

Furthermore, we can use out-of-domain data such as the texts on the WWW as an important information source. They have been successfully used to interpolate data sparseness in creating models for the word sense disambiguation (e.g., [90], [91], [92]) and can be helpful to refine WAMs in our methods.

On interesting research direction of QbT image retrieval may be the integration with auto-

matic image indexing. Automatic indexing is a classification task whose target task class has the size of a vocabulary. Statistically learned probabilistic models are now used for automatic image annotation [93], [94], [95]. In this thesis, we assumed that images are manually annotated and all annotations are correct. If annotations are provided automatically using heuristics or machine-learning techniques as referred to previously, the annotation will inevitably include many incorrect or ambiguous words since no automatic method has achieved human-like abilities to interpret the meanings of images. Researches on label noises in classification such as filtering noisy data [96] or incorporating label uncertainty [97] may serve as a guideline for future studies on annotation noises.

Table 3.5. This table shows the retrieval performances of the pLSI model and STM when increasing the size of annotations M from 1 to 3. The performances are measured by FPHR10, FPHR20, and the median in the category "Education."

$M = 1$

Method	FPHR10 (%)	FPHR20 (%)	Median (Position (%))
pLSI: Annotation and Image	5.80	9.18	106 (51.21)
STM: Annotation and Image	39.13	53.62	17 (8.21)

$M = 2$

Method	FPHR10 (%)	FPHR20 (%)	Median (Position (%))
pLSI: Annotation and Image	21.74	35.75	31 (14.98)
STM: Annotation and Image	51.21	69.57	10 (4.83)

$M = 3$

Method	FPHR10 (%)	FPHR20 (%)	Median (Position (%))
pLSI: Annotation and Image	30.92	53.14	19 (9.18)
STM: Annotation and Image	54.11	71.98	10 (4.83)

Chapter 4

Conclusions

4.1. Summary of Thesis

As demands to the intelligent information systems grow rapidly, probabilistic models tend to become more complex; therefore, more information is required to learn the models. This causes the data insufficiency problem. Focusing on two particular domains, classification and IR problems where the data insufficiency is crucial, this thesis has proposed novel methods to solve the problem of information insufficiency in statistical learning. The key technique is to use additional information sources. We have shown that the use of unlabeled sequences can help in the learning of HMMs and the use of paired images can advance cross-media image retrieval.

Conventionally, there have been some approaches involving the use of simplification methods, ensemble methods, and smoothing methods. These techniques leverage available information as much as possible but do not use any additional source. Through experimental results, we have found that the additional information usage brought remarkable improvements that have never been achieved without them.

4.2. Related Works and Future Directions

4.2.1 Interactive Learning

In both Chapter 2 and Chapter 3, we did not consider human interventions during learning. In the classification problem, observed measurements (feature vectors) do not contribute equally to construct class boundaries. That is, there are some informative data among them, and if we can select them and assign their class labels, we can obtain high-performance classifiers with little effort. This type of learning scheme is called *active learning*. In active learning, the system prompts the human labelers to assign class labels to the selected unlabeled data points.

The query-by-committee (QbC) algorithm is a typical active learning. QbC first designs

several different classifiers on the same labeled training data. It, then, selects unlabeled data points with strong disagreement among these classifiers' outputs and presents them to the human labelers, assuming that the stronger the disagreement is, the more the data are informative to the classifier design. The theoretical justification of QbC is given in [98]. The QbC algorithm has been applied to HMMs [99].

In IR, *relevance feedback* (RF), which has been discussed in Section 3.3.2, is an interactive learning scheme [100], [101]. In RF, a user is asked to assign relevance information to the initial IR output to refine the retrieval. In some domains, such as IR on the World Wide Web, interactivity sometimes does not work because relevance information is not obtainable; casual users are reluctant to be bothered by additional operations. In contrast, in some domains such as the analysis of scientific experiments, where experts' inspection of data has been the convention, even minimal reduction in human effort will be appreciated and experts may be willing to provide additional information. The use of interactivities in statistical learning should be examined from both theoretical and practical perspectives as above, and that should be a focus of our further research.

4.2.2 Computational Efficiency

We evaluated classification algorithms on the classification error rates and IR algorithms on the first-page hit rates. Computational efficiency was not a central issue in this thesis. As for classification, the proposed algorithm was designed based on the EM algorithm. Since various algorithms for speeding up the EM algorithm have been proposed (e.g. [102]), the learning speed of the proposed method can be improved by straightforwardly using those algorithms. In IR, retrieval speed is important. Currently, our IR method simply employs exhaustive search, although quick retrieval has become one of the most active areas in the database community (e.g. [103]). These methods developed there should be helpful in increasing the efficiency of our IR algorithm.

4.2.3 Other Information Sources

Other than the two information processing tasks we have explored, there may exist many applications where information for learning is quite insufficient, and these may suffer from various types of insufficiency. Also, within the scope of classification and IR, other types of information sources may be considered that have different types of heterogeneity. Viewed in this light, there are still various algorithms that need to be developed. One illustrative example is the use of pairwise constraints (e.g. two data must be in the same cluster or must not be in the same cluster) in clustering tasks. It has been reported that adding these constraints drastically improved clustering performance [104]. As another example, Cohn used similarity constraints in dimensionality reduction techniques so that the pair of data which belong to the same class should be projected to the same point in the reduced feature space [105].

4.2.4 Other Applications

In Chapter 2, we used HMMs for the classification and the learning algorithm was supervised. HMMs can also be used for the clustering of sequences (e.g. [40]) with unsupervised learning algorithms. If we apply our method to the clustering, we can perform *semi-supervised clustering*. In classification, we use labeled data as the basis of learning, with unlabeled data as an additional information source. In contrast, unlabeled data are the subject of the matter in clustering, whereas the label information is regarded as supplementary. One interesting aspect of semi-supervised clustering by using mixture models (mixture of HMMs in our case) is that we can find new classes which are not represented by labeled data. In this regard, we have carried out semi-supervised clustering with unknown classes [106]. A similar attempt has been made for the static data [107].

4.3. Final Remarks

Recently, we have been faced with flood of data; however, it is difficult to extract information from so much data. Ironically, we are still suffering from information insufficiency. The aim of this thesis has been to show the possibilities of using additional heterogeneous information sources that had previously been unused in traditional statistical learning settings.

We have shown two cases: one is the use of unlabeled sequences in learning HMM-based classifiers, the other is the use of paired data in learning STM-based cross-media image-retrieval systems. These are only two examples of such possibilities but we hope they highlight the potentialities of additional information. Furthermore, we hope that the methods developed in this thesis will stimulate future rigorous research on multi-source statistical learning.

References

[1] T.M. Mitchell, *Machine Learning*, Boston: The McGraw-Hill Companies, Inc., 1997.

[2] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford: Clarendon Press, 1995.

[3] I.T. Jolliffe, "Discarding Variables in Principal Component Analysis. I: Artificial Data," *Applied Statistics*, vol. 21, pp. 160–173, 1972.

[4] I.T. Jolliffe, "Discarding Variables in Principal Component Analysis. II: Real Data," *Applied Statistics*, vol. 22, pp. 21–31, 1973.

[5] S.C. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, and R.A. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society of Information Science*, vol. 41, no. 6, pp. 391-407, Sept. 1990.

[6] Y. Yang and J.O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," in *Proc. 14th Int'l Conf. Machine Learning*, pp. 412–420, 1997.

[7] W. Wang, P. Jones, and D. Partridge, "A Comparative Study of Feature-Salience Ranking Techniques," *Neural Computation*, vol. 13, no. 7, pp. 1603–1623, 2001.

[8] T.G. Dietterich, "Ensemble Methods in Machine Learning," in *First Int'l Workshop on Multiple Classifier Systems*, pp. 1–15, 2000.

[9] S.F. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," in *Proc. 34th Annual Meeting of the Association for Computational Linguistics*, pp. 310–318, 1996.

[10] C. Zhai and J. Lafferty, "A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval," in *Proc. 24th Annual Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 334–342, 2001.

[11] T.M. Mitchell, "Machine Learning and Data Mining," *Comm. ACM*, vol. 42, no. 11, pp. 30–36, Nov. 1999.

[12] G. Baliga, S. Jain, and A. Sharma, "Learning from Multiple Sources of Inaccurate Data," *SIAM J. Comput.*, vol. 26, no. 4, pp. 961–990, Aug. 1997.

[13] A.K. Jain, R.P.W Duin, and J. Mao, "Statistical Pattern Recognition: A Review," *IEEE Trans. on Pattern Anal. and Machine Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[14] T.G. Dietterich,R.H. Lathrop, and T. Lozano-Perez, "Solving the Multiple Instance Problem with Axis-Parallel Rectangles," *Artificial Intelligence*, vol. 89, no. 1-2, pp. 31–71, 1997.

[15] J. Bockhorst and M. Craven, "Exploiting Relations Among Concepts to Acquire Weakly Labeled Training Data," in *Proc. 19th Int'l Conf. Machine Learning*, pp. 43–50, 2002.

[16] V. Castelli and T.M. Cover, "On the Exponential Value of Labeled Samples," *Pattern Recognition Letters*, vol. 16, no. 1, pp. 105–111, 1995.

[17] V. Castelli and T.M. Cover, "The Relative Value of Labeled and Unlabeled Samples in Pattern Recognition with an Unknown Mixing Parameter," *IEEE Trans. Information Theory*, vol. 42, no. 6, pp. 2102–2117, 1996.

[18] T. Zhang and F. Oles, "A Probability Analysis on the Value of Unlabeled Data for Classification Problems," in *Proc. Int'l Conf. Machine Learning*, pp. 1191–1198, 2000.

[19] F.G. Cozman and I. Cohen, "Unlabeled Data Can Degrade Classification Performance of Generative Classifiers," in *15th Int'l Florida Artificial Intelligence Society Conf.*, pp. 327–331, 2002.

[20] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-training," in *Proc. Workshop on Computational Learning Theory*, pp. 92–100, 1998.

[21] T. Li, S. Zhu, Q. Li, and M. Ogihara, "Gene Functional Classification by Semi-supervised Learning from Heterogeneous Data," in *Proc. ACM Symposium on Applied Computing*, pp. 78–82, 2003.

[22] T. Kemp and A. Waibel, "Unsupervised Training of a Speech Recognizer: Recent Experiments," in *Proc. Eurospeech*, vol. 6, pp. 2725–2728, 1999.

[23] L. Lamel, J.L. Gauvain, and G. Adda, "Lightly Supervised and Unsupervised Acoustic Model Training," *Computer Speech and Language*, vol. 16, no. 1, pp. 115–129, Jan. 2002.

[24] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of Royal Statistical Society B*, vol. 39, no. 1, pp. 1-38, 1977.

[25] B.M. Shahshahani and D.A. Landgrebe, "The Effect of Unlabeled Samples in Reducing the Small Sample Size Problem and Mitigating the Hughes Phenomenon," *IEEE Trans. Geosci. and Remote Sensing*, vol. 32, no. 5, pp. 1087–1095, Sept. 1994.

[26] D.J. Miller and H.S. Uyar, "A Mixture of Experts Classifier with Learning Based on Both Labelled and Unlabelled Data," in *Advances in Neural Information Processing Systems 9*, pp. 571–577, 1997.

[27] K. Nigam, A.K. McCallum, S. Thrun, and T. Mitchell, "Text Classification from Labeled and Unlabeled Documents Using EM, " *Machine Learning*, vol. 39, no. 2/3, pp. 103–134, May 2000.

[28] S.M. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 35, no. 3, pp. 400–401, Mar. 1987.

[29] E.M. Voorhees, "Query Expansion using Lexical-Semantic Relations," in *Proc. 17th Annual Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 61–69, 1994.

[30] J. Gonzalo, F. Verdejo, I. Chugur and J. Cigarran, "Indexing with WordNet Synsets Can Improve Text Retrieval," in *Proc. COLING/ACL '98 Workshop on Usage of WordNet for NLP*, pp. 38–44, 1998.

[31] R. Ghani and R. Jones, "Learning a Monolingual Language Model from a Multilingual Text Database," in *Proc. 9th Int'l Conf. Information and Knowledge Management*, pp. 187–193, 2000.

[32] S. Sclaroff, M. La Cascia, S. Sethi, and L. Taycher, "Unifying Textual and Visual Cues for Content-Based Image Retrieval on the World Wide Web," *Computer Vision and Image Understanding*, vol. 75, no. 1-2, pp. 86–98, July 1999.

[33] A.F. Smeaton and I. Quigley, "Experiments on Using Semantic Distances Between Words in Image Caption Retrieval," in *Proc. 16th Annual Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 174–180, 1996.

[34] S. Flank, "A Layered Approach to NLP-Based Information Retrieval," in *Proc. 36th Annual Meeting of the Association for Computational Linguistics and 17th Int'l Conf. Computational Linguistics*, pp. 397–403, 1998.

[35] S.J. Raudys and A.K. Jain, "Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 13, no. 3, pp. 252–264, Mar. 1991.

[36] S. Ganesalingam and G.J. McLachlan, "Some Efficiency Results for the Estimation of the Mixing Proportion in a Mixture of Two Normal Distributions," Biometrics, vol. 37, pp. 22–33, March 1981.

[37] G.J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*, New York: John Wiley & Sons, 1992.

[38] L.R. Bahl, F. Jelinek, and R. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 5, pp. 179–190, Mar. 1983.

[39] T.E. Starner and A. Pentland, "Visual Recognition of American Sign Language Using Hidden Markov Models," in *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, pp. 189–194, 1995.

[40] A. Krogh, M. Brown, I.S. Mian, K. Sjölander, and D. Haussler, "Hidden Markov Models in Computational Biology Applications to Protein Modeling," *Journal of Molecular Biology*, vol. 235, no. 5, pp. 1501–1531, Feb. 1994.

[41] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun, "A Practical Part-of-Speech Tagger," in *Proc. Third Conf. Applied Natural Language Processing*, pp. 133–140, 1992.

[42] D.M. Bikel, R. Schwartz, and R.M. Weischedel, "An Algorithm that Learns What's in a Name," *Machine Learning*, vol. 34, no. 1-3, pp. 211–231, Feb. 1999.

[43] B. Merialdo, "Tagging English Text with a Probabilistic Model," *Computational Linguistics*, vol. 20, no. 2, pp. 155–171, June, 1994.

[44] D. Elworthy, "Does Baum-Welch Re-estimation Help Taggers?," in *4th Conf. Applied Natural Language Processing*, pp. 53–58, 1994.

[45] K. Seymore, A. McCallum, and R. Rosenfeld, "Learning Hidden Markov Model Structure for Information Extraction," in *AAAI Workshop on Machine Learning for Information Extraction*, pp. 37–42, 1999.

[46] J.R. Bellegarda and D. Nahamoo, "Tied Mixture Continuous Parameter Modeling for Speech Recognition," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 38, no. 12, pp. 2033–2045, Dec. 1990.

[47] X.D. Huang, "Phoneme Classification Using Semicontinuous Hidden Markov Models," *IEEE Trans. Signal Processing*, vol. 40, no. 5, pp. 1062–1067, May 1992.

[48] G. Aversano, A. Esposito, A. Esposito, and M. Marinaro, "A New Text-Independent Method for Phoneme Segmentation," in *Proc. 44th IEEE Midwest Symposium on Circuits and Systems*, vol. 2, pp. 516–519, 2001.

[49] L.E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Function of Markov Chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[50] X.D. Huang, Y. Ariki, and M.A. Jack, *Hidden Markov Models for Speech Recognition*, Edinburgh: Edinburgh University Press, 1990.

[51] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, 2nd ed., New York: John Wiley and Sons, 2001.

[52] F.H. Raab, E. Blood, T. Steiner, and H. Jones, "Magnetic Position and Orientation Tracking System," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 15, no. 5, pp. 709–717, Sept. 1979.

[53] J.S. Garofolo, L.F. Lamel, W.M. Fisher, J.G. Fiscus, D.S. Pallett, and N.L. Dahlgren, *DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CD-ROM*, Gaithersburg, MD: NIST, 1993.

[54] K.F. Lee and H.W. Hon, "Speaker-Independent Phone Recognition Using Hidden Markov Models," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, Nov. 1989.

[55] E. McDermott and S. Katagiri, "String-Level MCE for Continuous Phoneme Recognition," in *Proc. Eurospeech*, vol. 1, pp. 123–126, 1997.

[56] Y. Linde, A. Buzo, and R. Gray, "An Algorithm for Vector Quantizer Design," in *IEEE Trans. on Communications*, vol. 28, no. 1, pp. 84–95, Jan. 1980.

[57] G. Ball and D. Hall, "A Clustering Technique for Summarizing Multivariate Data," *Behavioral Science*, vol. 12, pp. 153–155, 1967.

[58] Y. Tsuruoka and J. Tsujii, "Training a Naive Bayes Classifier via the EM Algorithm with a Class Distribution Constraint," *Proc. CoNLL*, pp. 127–134, 2003.

[59] J. Larsen, A.S. Have, and L.K. Hansen, "Probabilistic Hierarchical Clustering with Labeled and Unlabeled Data," *Int'l Journal of Knowledge-Based Intelligent Engineering Systems*, vol. 6, no. 1, pp. 56–62, 2002.

[60] A. Corduneanu and T. Jaakkola, "Continuation Methods for Mixing Heterogeneous Sources," in *Uncertainty in Artificial Intelligence: Proc. 18th Conf.*, pp. 111–118, 2002.

[61] Z. Cataltepe and M. Magdon-Ismail, "Incorporating Test Inputs into Learning," in *Advances in Neural Information Processing Systems 10*, pp. 437–443, 1998.

[62] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A.W. Senior, "Recent Advances in the Automatic Recognition of Audiovisual Speech," *Proc. IEEE*, vol. 91, no. 9, pp. 1306–1326, Sept. 2003.

[63] R.A. Baeza-Yates and B.A. Ribeiro-Neto, *Modern Information Retrieval*, New York: ACM Press/Addison-Wesley, 1999.

[64] M.S. Lew and T.S. Huang, Visual Information Retrieval: Paradigms, Applications, and Research Issues, in *Principles of Visual Information Retrieval*, M.S. Lew Ed. London: Springer-Verlag, 2001 pp. 3–10.

[65] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-Based Image Retrieval at the End of the Early Years," *IEEE Trans. on Pattern Anal. and Machine Intell.*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000.

[66] D. Elworthy, "Retrieval from Captioned Image Databases using Natural Language Processing," in *Proc. 9th Int'l Conf. Information and Knowledge Management*, pp. 430–437, 2000.

[67] L. Khan and D. McLeod, "Effective Retrieval of Audio Information from Annotated Text Using Ontologies," in *Proc. Int'l Workshop on Multimedia Data Mining*, pp. 37–45, Aug. 2000.

[68] D. Hiemstra, "Using Language Models for Information Retrieval," *Ph.D. thesis, Centre for Telematics and Information Technology, University of Twente*, Jan., 2001.

[69] D.R. Miller, T. Leek, and R.M. Schwartz, "A Hidden Markov Model Information Retrieval System," in *Proc. 22th Annual Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 214–221, 1999.

[70] F. Rabitti and P. Savino, "An Information Retrieval Approach for Image Databases," in *Proc. 18th VLDB Conf.*, pp. 574–584, 1992.

[71] G.W. Furnas, T.K. Landauer, L.M. Gomez, and S.T. Dumais, "The Vocabulary Problem in Human-System Communication," *Comm. ACM*, vol. 30, no. 11, pp. 964–971, Nov. 1987.

[72] Y. Qiu and H.P. Frei, "Concept-Based Query Expansion," in *Proc. 16th Annual Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 160–169, 1993.

[73] J. Xu and W.B. Croft, "Query Expansion Using Local and Global Document Analysis," in *Proc. 19th Annual Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 4–11, 1996.

[74] G. Salton and C. Buckley, "Improving Retrieval Performance by Relevance Feedback," *Journal of the American Society for Information Science*, vol. 41, no. 4, pp. 288-297, June 1990.

[75] C. Buckley, G. Salton, J. Allan, and A. Singhal, "Automatic Query Expansion Using SMART: TREC 3," in *Proc. Third Text REtrieval Conf. (TREC-3)*, pp. 69–80, 1994.

[76] Y. Lu, H. Zhang, L Wenyin, and C. Hu, "Joint Semantics and Feature Based Image Retrieval using Relevance Feedback," *IEEE Trans. on Multimedia*, vol. 5, no. 3, pp. 339–347, Sept. 2003.

[77] X.S. Zhou and T.S. Huang, "Unifying Keywords and Visual Contents in Image Retrieval," *IEEE Multimedia*, vol. 9, no. 2, pp. 23–33, April-June, 2002.

[78] G.A. Miller, "WordNet: a Lexical Database for English," *Comm. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995.

[79] H. Schütze and J.O. Pederson, "A Cooccurrence-Based Thesaurus and Two Applications to Information Retrieval," *Information Processing and Management*, vol. 33, no. 3, pp. 307–318, 1997.

[80] A. Berger, "Statistical Machine Learning for Information Retrieval," *Ph.D. thesis, School of Computer Science, Carnegie Mellon University*, CMU-CS-01-110, 2001.

[81] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine Learning*, vol. 42, no. 1, pp. 177–196, 2001.

[82] L. Azzopardi, M. Girolami, and K. van Risjbergen, "Investigating the Relationship between Language Model Perplexity and IR Precision-Recall Measure," in *Proc. 26th Annual Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, 2003.

[83] R. Zhao and W.I. Grosky, "Narrowing the Semantic Gap - Improved Text-Based Web Document Retrieval Using Visual Features," *IEEE Trans. on Multimedia*, vol. 4, no. 2, pp. 189–200, June, 2002.

[84] E. Brochu, N. de Freitas, and K. Bao, "The Sound of an Album Cover: Probabilistic Multimedia and IR," in *9th Annual Workshop on Artificial Intelligence and Statistics*, 2003.

[85] H. Müller, W. Müller, S. Marchand-Maillet, T. Pun, and D.McG. Squire, "A Framework for Benchmarking in CBIR," *Int'l Journal on Multimedia Tools and Applications*, vol. 21, no. 2, pp. 55–73, 2003.

[86] A. Rao, R.K. Srihari, Z. Lei, and A. Zhang, "A Method for Measuring the Complexity of Image Databases," *IEEE Trans. on Multimedia*, vol. 4, no. 2, pp. 160–173, June 2002.

[87] B.J. Jansen, A. Spink, and T. Saracevic, "Real Life, Real Users, and Real Needs: a Study and Analysis of User Queries on the Web," *Information Processing and Management*, vol. 36, no. 2, pp. 207–227, 2000.

[88] M. Mitra, A. Singhal, and C. Buckley, "Improving Automatic Query Expansion," in *Proc. 21th Annual Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 206–214, 1998.

[89] J. Maintz and M. Viergever, "A Survey of Medical Image Registration," *Medical Image Analysis*, vol. 2, no. 1, pp. 1–36, 1998.

[90] R. Mihalcea and D. Moldovan, "A Method for Word Sense Disambiguation of Unrestricted Text," in *Proc. 37th Annual Meeting of the Association for Computational Linguistics*, pp. 152–158, 1999.

[91] E. Agirre, O. Ansa, E. Hovy, and D. Martinez, "Enriching Very Large Ontologies Using the WWW," in *Proc. Ontology Learning Workshop, ECAI*, 2000.

[92] F. Keller and M. Lapata, "Using the Web to Obtain Frequencies for Unseen Bigrams," *Computational Linguistics*, vol. 29, no. 3, pp. 459–484, 2003.

[93] Y. Mori, H. Takahashi, and R. Oka, "Image-to-Word Transformation Based on Dividing and Vector Quantizing Images with Dords," in *First Int'l Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999.

[94] J. Jeon, V. Lavrenko, and R. Manmatha, "Automatic Image Annotation and Retrieval using Cross-Media Relevance Models," in *Proc. 26th Annual Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 119–126, 2003.

[95] J. Li and J. Z. Wang, "Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach," *IEEE Trans. on Pattern Anal. and Machine Intell.*, vol. 25, no. 9, pp. 1075–1088, Sept. 2003.

[96] C.D. Brodley and M.A. Friedl, "Identifying Mislabeled Training Data," *Journal of Artificial Intelligent Research*, vol. 11, pp. 131–167, 1999.

[97] N. D. Lawrence and B. Schölkopf, "Estimating a Kernel Fisher Discriminant in the Presence of Label Noise," in *Int'l Conf. Machine Learning*, pp. 306–313, 2001.

[98] H.S. Seung, M. Opper, and H. Sompolinsky, "Query by Committee," in *Computational Learning Theory*, pp. 287–294, 1992.

[99] S. Argamon-E and I. Dagan, "Committee-Based Sample Selection for Probabilistic Classifiers," *Journal of Artificial Intelligence Research*, vol. 11, pp. 335–360, 1999.

[100] D.D. Lewis and W.A. Gale, "A Sequential Algorithm for Training Text Classifiers," in *Proc. 17th Annual Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 3–12, 1994.

[101] D.D. Lewis, "A Sequential Algorithm for Training Text Classifiers: Corrigendum and Additional Data," *SIGIR Forum*, vol. 29, no. 2, pp. 13–19, 1995.

[102] B. Thiesson, C. Meek, and D. Heckerman, "Accelerating EM for Large Databases," *Machine Learning*, vol. 45, no. 3, pp. 279–299, 2001.

[103] C. Faloutsos and K.I. Lin, "FastMap: a Fast Algorithm for Indexing, Data-mining and Visualization of Traditional and Multimedia Datasets," in *Proc. ACM SIGMOD Int'l Conf. Management of Data*, pp. 163–174, 1995.

[104] D. Klein, S.D. Kamvar, and C.D. Manning, "From Instance-level Constraints to Space-Level Constraints: Making the Most of Prior Knowledge in Data Clustering," in *Proc. 19th Int'l Conf. Machine Learning*, pp. 307–314, 2002.

[105] D. Cohn, "Informed Projections," *Advances in Neural Information Processing Systems 15*, pp. 849–856, 2003.

[106] N. Ueda and M. Inoue, "Clustering Labeled and Unlabeled Temporal Gene Expression Profiles," in *Workshop on Information-Based Induction Sciences*, pp. 139–146, 2002 (in Japanese).

[107] D.J. Miller and J. Browning, "A Mixture Model and EM-based Algorithm for Class Discovery, Robust Classification, and Outlier Rejection in Mixed Labeled/Unlabeled Data Sets," *IEEE Trans. on Pattern Anal. and Machine Intell.*, vol. 25, no. 11, pp. 1468–1483, Nov. 2003.

[108] R.A. Jacobs, M.I. Jordon, S.J. Nowlan, and G.E. Hinton, "Adaptive Mixtures of Local Experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.

[109] V. Pavlovic, R. Sharma, and T. Huang, "Visual Interpretation of Hand Gestures for Human-computer Interaction: A Review," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 19, no. 7, pp. 677–695, July 1997.

Appendix

A. Choice of Probabilistic Models

ETM-HMMs are not the only ways to incorporate unlabeled data directly into the training of HMMs. Another possibility is an extension of the static model based on the mixture of experts (MoE) framework [108]. This method has been used by Miller and Uyar [26]. In contrast to the MoE framework, we call our approach the tied-mixture (TM) framework.

On one hand, the probabilistic model of an ETM-HMM, a mixture of HMMs within our TM framework, is written as follows:

$$\begin{aligned} & P(X|\Theta) \\ = & \sum_y P(y) \sum_S \sum_M p(X, S, M|y, \Theta) \\ = & \sum_Y P(y) \sum_S \sum_M P(S|y, \Theta) P(M|S, y, \Theta) p(X|S, M, y, \Theta) \\ \approx & \sum_Y P(y) \sum_S \sum_M P(S|y, \Theta) P(M|S, \Theta) p(X|S, M, y, \Theta) \\ \approx & \sum_Y P(y) \sum_S \sum_M P(S|y, \Theta) P(M|S, \Theta) p(X|M, \Theta), \end{aligned} \tag{4.1}$$

where $P(y)$ is the class prior, $P(S|y, \Theta)$ is the state transition probability, $P(M|S, \Theta)$ is the state-conditional (i.e., class conditional) mixture coefficient, and $p(X|M, \Theta)$ is the distribution represented by a Gaussian. The last transformation means Gaussians are tied over classes. The second-last transformation means that the mixture coefficient depends on y, not directly but indirectly, through S, which depends on y.

On the other hand, the probabilistic model of the mixture of HMMs within the MoE framework is written as follows:

$$\begin{aligned} & P(X|\Theta) \\ = & \sum_y \sum_S \sum_M p(X, y, S, M|\Theta) \\ = & \sum_y \sum_S \sum_M P(y|S, M, X, \Theta) P(S, M|\Theta) p(X|S, M, \Theta) \\ \approx & \sum_y \sum_S \sum_M P(y|S, M, X, \Theta) P(S|\Theta) P(M|S, \Theta) p(X|M, \Theta) \\ \approx & \sum_y \sum_S \sum_M P(y|S, M, \Theta) P(S|\Theta) P(M|S, \Theta) p(X|M, \Theta), \end{aligned} \tag{4.2}$$

where $P(y|S, M, \Theta)$ is the stochastic class selector, $P(S|\Theta)P(M|S, \Theta)$ is the gating function, and $p(X|M, \Theta)$ is the local committee (expert) represented by a Gaussian. The last transformation assumes the independence of the class selectors from feature vectors. The second-last transformation means that the output depends only on M and not on S.

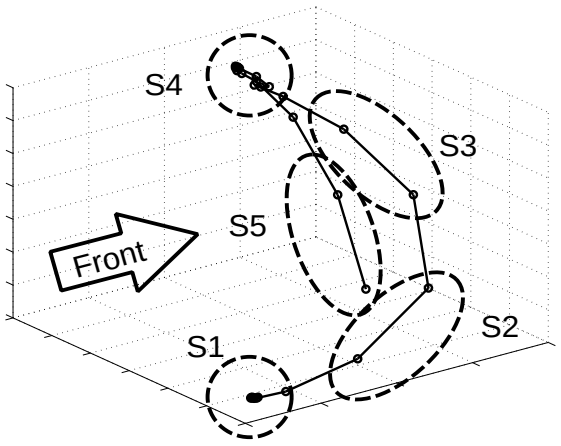


Figure 4.1. A trajectory of the right hand while signing "aisatsu" in JSL, which consists of 29 sampling points. Its five states are conceptualized by S1,$\cdots$,S5. The first state corresponds to the initial position of the hand (around the chest). In the second state, the hand is pushed forward. In the third state, it is raised. In the fourth state, it stays in front of the face, and in the fifth state, it returns to the initial position.

In the TM framework given by (4.1), the primitives of phenomena (or state) remain interpretable; $P(S|y,\Theta)$ in (4.1) corresponds to a particular stationary process of class y phenomenon. For example, a sign in sign language is viewed as a sequence of primitive hand movements (class-dependent state sequences). An example of such primitives of Japanese Sign Language (JSL) signs used in our experiment in Section 2 is shown in Fig. 4.1. This interpretability of the states is sometimes considered to be an interesting feature of HMMs in an application such as gesture understanding [109]. In contrast, $P(S|\Theta)$ in (4.2) is difficult to interpret since it is mixed over different classes. We choose to use the TM model because of this differences.

B. Scaling for the unlabeled posteriors

This appendix describes the scaling technique we used to prevent the computational problem that occurs during the calculation of unlabeled posteriors. In HMMs, posteriors are efficiently calculated by the forward-backward algorithm [50]. However, if the observed sequence is long, the values of the intermediate variables used in the forward-backward algorithm become too small to be handled within the precision range of computers. For labeled data, this problem can be avoided by applying a scaling technique [50]. For unlabeled data, however, it is not applicable. In the following, we show why conventional scaling does not work and present a modified scaling procedure that is applicable to unlabeled data.

As an example, we consider the calculation of transition posteriors for labeled or unlabeled data (2.13) or (2.17). Here and in the following, for simplicity of notation, we assume a single observation and omit the index n. Let $\alpha_t^y(i) = p(\mathbf{x}_1, ..., \mathbf{x}_t, s_t = i|\ y, \Theta)$ be the forward variable unscaled and computed from time 1 to t in the state i at time t. Let $\beta_t^y(j) = p(\mathbf{x}_{t+1}, ..., \mathbf{x}_T|\ s_t = i, y, \Theta)$ be the backward variable unscaled and computed from time T to t in the state j at time t. Let $\tilde{\alpha}_t^y(i)$ be the forward variable scaled from time 1 to time $t-1$ in the state i at time t, and $\tilde{\beta}_t^y(j)$ be the backward variable scaled from time T to time $t+1$ in the state j at time t. Let $\mathrm{W}_t^y = 1/\sum_i \tilde{\alpha}_t^y(i)$ be the scaling coefficient at time t, $\check{\alpha}_t^y(i) = \mathrm{W}_t^y \cdot \tilde{\alpha}_t^y(i)$ be the forward variable scaled from time 1 to time t, and $\check{\beta}_t^y(j) = \mathrm{W}_t^y \cdot \tilde{\beta}_t^y(j)$ be the backward variables scaled from time T to time t. Between scaled forward or backward variables, unscaled forward or backward variables, and scaling coefficients, the following relationships holds [50]:

$$\check{\alpha}_t^y(i) = \left[\prod_{t'=1}^{t} \mathrm{W}_{t'}^y\right] \cdot \alpha_t^y(i), \tag{4.3}$$

$$\check{\beta}_{t+1}^y(j) = \left[\prod_{t'=t+1}^{T} \mathrm{W}_{t'}^y\right] \cdot \beta_{t+1}^y(j). \tag{4.4}$$

For labeled data, if this scaling procedure is applied, since the scaling coefficients in the numerator and the denominator cancel out, $\gamma_t^y(i,j)$ can be calculated by the following equation:

$$\gamma_t^y(i,j) = \frac{\check{\alpha}_t^y(i) a_{ij}^y b_j^y(\mathbf{x}_{t+1}) \check{\beta}_{t+1}^y(j)}{\sum_{i \in F^{y'}} \check{\alpha}_T^{y'}(i)}, \tag{4.5}$$

where $b_j^y(\mathbf{x}_{t+1}) = \sum_k c_{jk}^y \mathcal{N}(\mathbf{x}_{t+1}|\mu_k, \Sigma_k)$ and F^y represents the final state among the states of the class y HMM. For unlabeled data, if the scaling procedure is applied, $\lambda_t(y,i,j)$ may be calculated by the following equation:

$$\lambda_t(y,i,j) = \frac{\omega_y \check{\alpha}_t^y(i) a_{ij}^y b_j^y(\mathbf{x}_{t+1}) \check{\beta}_{t+1}^y(j)}{\left[\prod_{t'=1}^{T} \mathrm{W}_{t'}^y\right] \sum_{y'} \omega_{y'} \sum_{i \in F^{y'}} \alpha_T^{y'}(i)}. \tag{4.6}$$

In (4.6), with the help of the scaling, each variable in the numerator can be calculated at each time t; while, in the denominator, the product of scaling coefficients $\prod_{t'=1}^{T} \mathrm{W}_{t'}^y$ cannot be calculated when the length of the sequence T is large. This is because W_t^y is usually large

at each t and the calculation of the product often reaches infinity on computers. To avoid this problem, the following successive computation procedure is introduced. We transform the denominator of (4.6) as follows:

$$\begin{aligned} & \left[\prod_{t'=1}^{T} \mathrm{W}_{t'}^{y}\right] \sum_{y'} \omega_{y'} \sum_{i \in F^{y'}} \alpha_{T}^{y'}(i) \\ = & \prod_{t'=1}^{T} \mathrm{W}_{t'}^{y} \sum_{y'} \omega_{y'} \frac{1}{\prod_{t'=1}^{T} \mathrm{W}_{t'}^{y'}} \\ = & \sum_{y'} \omega_{y'} \prod_{t'=1}^{T} \left(\frac{\mathrm{W}_{t'}^{y}}{\mathrm{W}_{t'}^{y'}}\right). \end{aligned} \tag{4.7}$$

By so doing, in most practical cases, the value of $\mathrm{W}_{t}^{y}/\mathrm{W}_{t}^{y'}$ remains computable. Other posteriors for unlabeled data can be calculated by applying this transformation.

C. Convergence Criterion

This appendix describes the convergence criterion used in this paper. When the increase in likelihood is smaller than a predefined threshold, we regard the EBW algorithm to be converged. This is possible on condition that the likelihood, or log-likelihood, is computable. The log-likelihood of an ETM-HMM is given as follows:

$$\begin{aligned}\mathcal{L}(\Theta|\mathcal{D}) &= \mathcal{L}(\Theta|\mathcal{D}_l) + \mathcal{L}(\Theta|\mathcal{D}_u) \\ &= \sum_{y=1}^{Y}\sum_{n\in\mathcal{I}_y}\log p(X_n, y_n|\Theta) + \sum_{n=1}^{N_u}\log p(X_n|\Theta) \\ &= \sum_{y=1}^{Y}\sum_{n\in\mathcal{I}_y}\log \omega_y p(X_n|y,\Theta) + \sum_{n=1}^{N_u}\log\sum_{y}\omega_y p(X_n|y,\Theta). \end{aligned} \tag{4.8}$$

When scaling coefficients $\mathrm{W}_{n_t}^{y}$ defined in Appendix B are used, since $\prod_{t=1}^{T_n}\mathrm{W}_{n_t}^{y} = p(X_n|y,\Theta)$ holds, (4.8) is rewritten as:

$$\mathcal{L}(\Theta|\mathcal{D}) = \sum_{y=1}^{Y}\sum_{n\in\mathcal{I}_y}\log\frac{\omega_y}{\prod_{t=1}^{T_n}\mathrm{W}_{n_t}^{y}} + \sum_{n=1}^{N_u}\log\sum_{y=1}^{Y}\frac{\omega_y}{\prod_{t=1}^{T_n}\mathrm{W}_{n_t}^{y}}. \tag{4.9}$$

As has been explained in Appendix B, $\prod_{t=1}^{T_n}\mathrm{W}_{n_t}^{y}$ cannot be computed when T_n is large. To make the log-likelihood computable, we introduce a meta-scaling coefficient V whose value is defined empirically according to the data. Let N be the sum of N_l and N_u. By substituting the product of V from (4.9), we have:

$$\begin{aligned}\mathcal{L}'(\Theta|\mathcal{D}) &= \mathcal{L}(\Theta|\mathcal{D}) - \log\prod_{n=1}^{N}\prod_{t=1}^{T_n}\mathrm{V} \\ &= \mathcal{L}(\Theta|\mathcal{D}) - \sum_{y=1}^{Y}\sum_{n\in\mathcal{I}_y}\log\prod_{t=1}^{T_n}\mathrm{V} - \sum_{n=1}^{N_u}\log\prod_{t=1}^{T_n}\mathrm{V} \\ &= \sum_{y=1}^{Y}\sum_{n\in\mathcal{I}_y}\log\frac{\omega_y}{\prod_{t=1}^{T_n}(\mathrm{V}\cdot\mathrm{W}_{n_t}^{y})} + \sum_{n=1}^{N_u}\log\sum_{y=1}^{Y}\frac{\omega_y}{\prod_{t=1}^{T_n}(\mathrm{V}\cdot\mathrm{W}_{n_t}^{y})}. \end{aligned} \tag{4.10}$$

Thus, if we choose V appropriately so that $\prod_{t=1}^{T}(\mathrm{V}\cdot\mathrm{W}_{n_t}^{y})$ is computable, we can obtain $\mathcal{L}'(\Theta|\mathcal{D})$. Although $\mathcal{L}'(\Theta|\mathcal{D})$ is no longer the log-likelihood itself, since the substituted value, $\log\prod_{n=1}^{N}\prod_{t=1}^{T_n}\mathrm{V}$, is a constant, we can use the change in $\mathcal{L}'(\Theta|\mathcal{D})$ to determine the convergence of the EBW algorithm.

D. Class Distribution of TIMIT Data

This appendix shows the data distributions in 48 classes of TIMIT data used in Section 2.6.3. In contrast to the JSL data used in Section 2.6.2, the quantities of data in each class of TIMIT are diverse. Table 4.1 shows the distribution for the training data and Table 4.2 shows that for the test data.

Table 4.1. The distribution of TIMIT training data.

Phone	No. of Sequences	Percentage(%)	Phone	No. of Sequences	Percentage(%)
aa	2,252	1.61	iy	4,598	3.28
ae	2,292	1.64	jh	1,013	0.72
ah	2,623	1.87	k	3,794	2.71
ao	1,861	1.33	l	4,423	3.16
aw	728	0.52	m	3,566	2.55
ax	3,534	2.52	n	6,894	4.92
ay	1,917	1.37	ng	1,220	0.87
b	2,181	1.56	ow	1,642	1.17
ch	820	0.59	oy	304	0.27
cl	12,516	8.93	p	2,588	1.85
d	2,432	1.74	r	4,680	3.34
dh	2,376	1.70	s	6,176	4.41
dx	1,864	1.33	sh	1,317	0.94
eh	3,275	2.34	sil	8,282	5.91
el	951	0.68	t	3,948	2.82
en	630	0.45	th	745	0.53
epi	908	0.65	uh	500	0.36
er	4,108	2.93	uw	1,947	1.39
ey	2,265	1.62	v	1,994	1.42
f	2,215	1.58	vcl	7,217	5.15
g	1,191	0.85	w	2,216	1.58
hh	1,659	1.18	y	995	0.71
ih	4,245	3.03	z	3,682	2.63
ix	7,366	5.26	zh	149	0.11
			total	140,099	100

Table 4.2. The distribution of TIMIT test data.

Phone	No. of Sequences	Percentage(%)	Phone	No. of Sequences	Percentage(%)
aa	846	1.67	iy	1,810	3.57
ae	772	1.52	jh	295	0.58
ah	974	1.92	k	1,204	2.37
ao	761	1.50	l	1,858	3.66
aw	216	0.43	m	1,406	2.77
ax	1,321	2.60	n	2,434	4.80
ay	686	1.35	ng	378	0.74
b	886	1.75	ow	600	1.18
ch	259	0.51	oy	127	0.25
cl	4,300	8.47	p	957	1.89
d	841	1.66	r	1,849	3.64
dh	896	1.77	s	2,172	4.28
dx	634	1.25	sh	460	0.91
eh	1,247	2.46	sil	3,053	6.02
el	343	0.68	t	1,367	2.69
en	216	0.43	th	259	0.51
epi	332	0.65	uh	215	0.42
er	1,692	3.33	uw	572	1.13
ey	802	1.58	v	710	1.40
f	911	1.79	vcl	2,553	5.03
g	452	0.89	w	903	1.78
hh	561	1.11	y	376	0.74
ih	1,438	2.83	z	1,236	2.44
ix	2,501	4.93	zh	73	0.144
			total	50,754	100

List of Publications

Journal Papers

- INOUE, Masashi and UEDA, Naonori, "Exploitation of Unlabeled Sequences in Hidden Markov Models." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 11, pp. 1570–1581, Dec. 2003.
- UEDA, Naonori and INOUE, Masashi, "Extended Tied-Mixture HMMs for both Labeled and Unlabeled Time Series Data," *Journal of VLSI Signal Processing Systems*, to appear.
- INOUE, Masashi and UEDA Naonori, "Use of Unlabeled Time Series Data in Hidden Markov Models" IEICE Trans., Vol. J86-D-II, No. 2, pp. 173–183, Feb. 2003 (in Japanese).

International Conferences and Workshops

- INOUE, Masashi and UEDA, Naonori, "HMMs for both Labeled and Unlabeled Time Series Data," 11th IEEE workshops on Neural Networks for Signal Processing, pp. 93–102, Sept. 2001.

Other Publications

- INOUE, Masashi and UEDA Naonori, "Use of Unlabeled Time Series Data in Hidden Markov Models," Systems and Computers in Japan Vol. 34, No. 13, pp. 1–12, Nov. 2003.
- UEDA, Naonori and INOUE, Masashi, "Clustering Labeled and Unlabeled Temporal Gene Expression Profiles," Workshop on Information-Based Induction Sciences, pp. 139–146, Sep. 2002 (in Japanese).
- INOUE, Masashi and UEDA Naonori, "Recognition Method using Both Labeled and Unlabeled Time Series Data," IEICE Tech. Rep., NC2000-129, pp. 9–16 March, 2000 (in Japanese).

Others

- INOUE, Masashi, HIGUCHI, Satomi, KAWAWAKI, Dai, KANEKO, Yuko, and NIKI, Chiharu, "Reading of Handwritten and Printed Text: An fMRI Study," The Ninth Annual Meeting of the Organization for Human Brain Mapping, abstract No. 1296, June 2003.