A DETECTION-BASED PATTERN RECOGNITION FRAMEWORK AND ITS APPLICATIONS

A Thesis
Presented to
The Academic Faculty

by

Chengyuan Ma

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2010

A DETECTION-BASED PATTERN RECOGNITION FRAMEWORK AND ITS APPLICATIONS

Approved by:

Prof. Chin-Hui Lee, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Prof. Mark Clements
School of Electrical and Computer Engineering
Georgia Institute of Technology

Prof. Justin Romberg
School of Electrical and Computer Engineering
Georgia Institute of Technology

Prof. Maysam Ghovanloo
School of Electrical and Computer Engineering
Georgia Institute of Technology

Prof. Ming Yuan
School of ISyE
Georgia Institute of Technology

Date Approved: April 2, 2010

To my family,

Yonghu Ma, Guixiang Yan, Yunzhe Zhang, and Jianwei Ma

ACKNOWLEDGEMENTS

I would like to express my profound gratitude to my advisor, Prof. Chin-Hui Lee, for his guidance and support throughout my PhD study at Georgia Tech. This work would not have been possible without his encouragement and support during the last several years. His insights and vision on many research topics have directed my research work to a deeper level and broader view. I am so lucky to have been working with Dr. Lee in such an important period of my life. His guidance has shaped many aspects of my research and communication skills.

I would also like to thank Prof. Mark Clements, I have learnt a lot from him in our joint ASAT project. I also greatly appreciate Prof. Justin Romberg, Prof. Maysam Ghovanloo, and Prof. Ming Yuan for serving on my thesis committee.

I owe great acknowledgements to many researchers in IBM T.J. Watson research center, Dr. Jeff Kuo, Dr. Lidia Mangu, Dr. Hagen Soltau, Dr. Xiaodong Cui, Dr. Upendra Chaudhari, and many researchers in Microsoft research, Dr. Patrick Nguyen, Dr. Milind Mahajan, Dr. Li Deng, whose precious advices and instructions helped me having a productive summer internship.

I am also very grateful to many friends at Georgia Tech: Jian Zhu, Jinyu Li, Yu Tsao, Qiang Fu, Yong Zhao, You-chi Cheng, Antonio Moreno, Brett Matthews, Yeongseon Lee, Jeremy Reed, Sibel Yaman, Sabato Marco Siniscalchi, Byungki Byun, Ilseo Kim, Xusheng Sun, Junlin Li, Wei Zhang, Kun Shi, and Zhensheng Jia, for their collaboration, help, and fun time.

Finally, I would like to express my deepest gratitude to my parents, my wife, and my son for their great love and consistent support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The objective of this dissertation is to present a detection-based pattern recognition framework and demonstrate its applications in automatic speech recognition and broadcast news video story segmentation.

Inspired by the studies of modern cognitive psychology and real-world pattern recognition systems, a detection-based pattern recognition framework is proposed to provide an alternative solution for some complicated pattern recognition problems. The primitive features are first detected and the task-specific knowledge hierarchy is constructed level by level. Then, a variety of heterogeneous information sources are combined together and the high level context is incorporated as additional information at certain stages.

A detection-based framework is a "divide-and-conquer" design paradigm for pattern recognition problems, which will decomposes a conceptually difficult problem into many elementary subproblems that can be handled directly and reliably. Some information fusion strategies will be employed to integrate the evidence from a lower level to form the evidence at a higher level. Such a fusion procedure continues until reaching the top level. Generally, a detection-based framework has many advantages: (1) more flexibility in both detector design and fusion strategies, as these two parts can be optimized separately; (2) parallel and distributed computational components in primitive feature detection. In such a component-based framework, any primitive component can be replaced by a new one while other components remain unchanged; (3) incremental information integration; (4) high level context information as additional information sources, which can be combined with bottom-up processing at any

stage.

This dissertation presents the basic principles, criteria, and techniques for detector design and hypothesis verification based on the statistical detection and decision theory. In addition, evidence fusion strategies were investigated in this dissertation. Several novel detection algorithms and evidence fusion methods were proposed and their effectiveness was justified in automatic speech recognition and broadcast news video segmentation system. We believe such a detection-based framework can be employed in more applications in the future.

CHAPTER I

INTRODUCTION

Pattern recognition is a complicated procedure of information acquisition and processing to correctly perceive, describe, categorize, and interpret ambiguous sensory information. It is an elementary yet remarkable internal mental process of human beings. Cognitive psychologists and computer scientists alike have been conducting research on pattern recognition over the last several decades but from different perspectives.

Computer scientists have invented numerous computational models and algorithms to mimic human beings' pattern recognition capabilities and make computers think more like human beings. These models and techniques have been employed in many real-world applications such as automatic speech recognition (ASR), video analysis, text categorization (e.g., spam/non-spam emails), automatic handwriting recognition, and face recognition, just to name several examples.

For cognitive psychologists, the study of pattern recognition is of great importance because pattern recognition is a critical part of human perception. The core focus of cognitive psychology is on how human beings acquire, process, and store information. Cognitive psychology research has been unified by a common approach based on an analogy between the human mind and computers, which is the information-processing approach in modern cognitive psychology [25]. Many cognitive psychologists treat people as dynamic information-processing systems whose mental operations might be described in computational terms.

To leverage the studies of cognitive psychology in practical pattern recognition systems, it is worth reviewing the research on human perception from a cognitive

psychologist's perspective.

1.1 Top-down and Bottom-up Perception

Studies on human perception focus on how human beings take stimuli from the environment and convert it into a representation that the mind can use. Many cognitive psychologists think that two major classes of approaches are generally used in human perception: top-down and bottom-up processing [25].

Top-down processing refers to expectation-driven perception, which posits varying degrees of influence of higher cognitive processes on what we actually perceive. Our perceptual experience is influenced by higher level cognitive processes, such as expectation, knowledge, context, experience, and thoughts. Human perception is not determined simply by stimulus patterns; rather it is a dynamic searching for the best interpretation of the available data [25].

Bottom-up processing refers to stimulus-driven perception. In other words, the physical properties of a stimulus (e.g., color, pitch, motion) can influence the perception of a given stimulus. Human perception builds up hierarchically from a set of primitive features to our internal representations. All bottom-up theories rely on the notion that perception builds upward from a foundation of primitives to a representation our cognitive system can use. It takes place without any influence from higher cognitive processes. There are several theories of bottom-up perception in cognitive psychology. The following description is a brief summary of the major bottom-up theories [25]. According to this viewpoint, information processing typically proceeds through two stages: (1) specific forms of processing in several brain subsystems; (2) integration of information from these brain subsystems.

Direct perception Perception is a direct result of stimulus energy affecting receptor cells. No higher cognitive processes or internal representations are necessary.

Template/Exemplar theory Examples of all the objects we have seen are stored

as exemplars or templates in our mind. We compare a perceived object to this set of exemplars until we find a match.

Prototype theory Instead of storing many exemplars or rigid templates, we store a prototype, which is an average of objects in some sense. We compare a perceived object to these prototypes until we find the closest match.

Feature theory Perception starts with the identification of basic features, such as lines and corners, that are then put together into more complex objects until we identify an object.

Top-down and bottom-up processing can be summarized as follows: bottom-up (or stimulus-driven processing) is directly affected by stimulus input, whereas top-down (or expectation-driven processing) is directly affected by context and past experience. As an example of top-down processing, it is easier to identify the word "wheel" in poor handwriting if it is presented in a sentence context, "The *eel was on the axle" [25], than when it is presented on its own. This indicates the importance of context for human perception and how it can greatly improve the robustness of perception.

Other studies from computer vision and cognitive psychology show that human vision perception often occurs in a bottom-up style [109]. The study of vision, in both humans and machines, can be viewed as the discovery of constraints. As additional constraints are added, the hypotheses become sharper and more focused, particularly when the constraints span several descriptive levels. Enough constraints will become known finally at each level so that a hypothesis will become the correct one [28] [109]. As Gazzaniga stated in [36], "Visual perception is a divide-and-conquer strategy. Rather than have each visual area represent all attributes of an object, each area provides its own limited analysis. So, the processing is distributed and specialized." This visual perception model has provided the motivation for the investigation of component-based detection algorithms [2] [18] [40] [102], where the detection of the

entire object is through an integration of the detection of its parts; the different features of an object are probably integrated in a subsequent higher integrative cortical area. This approach may provide a powerful means to overcome the limitation of the top-down object detectors, because the variability of the target object is decomposed into the local variability of its individual parts.

There is evidence to support the concept that human speech recognition (HSR) involves a hierarchical bottom-up analysis [3]. For instance, people do not continuously convert speech signals into words as an ASR system attempts to do. Instead, they detect acoustic and auditory cues, combine them to form cognitive hypotheses, and then validate the hypotheses until consistent decisions are obtained. Then, multiple knowledge sources are integrated into the recognition process [59]. Human speech recognition involves a bottom-up, divide-and-conquer strategy. We recognize speech based on a hierarchy of context layers. As in vision, entropy decreases as we integrate context.

Other evidence demonstrates that human speech recognition is achieved by a mixture of bottom-up processing triggered by acoustic signals and top-down processing generated from linguistic contexts. However, there have been disagreements about precisely how information from bottom-up and top-down processes is combined to produce word recognition [25].

The notion that bottom-up processing in HSR makes use of lower level distinctive feature information was supported in a classic study by Miller and Nicely [79]. They gave their participants the task of recognizing consonants against background noises. The most frequently confused consonants were those differing on the basis of only one distinctive feature, which means if we can detect these distinctive features accurately, these consonants could be correctly identified [25].

Evidence that top-down processing based on context is involved in speech perception was obtained by Warren and Warren [104]. In their experiments, participants

heard a sentence in which a small portion had been removed and replaced with a meaningless sound. The auditory stimulus was always the same, so all that differed was the contextual information. They concluded that the perception of the crucial element in the sentence was influenced by sentence context.

In a later study by Samuel [25], he identified two possible explanations for these experimental results. First, context may interact directly with bottom-up processes. Second, the context may simply provide an additional source of information. Performance in Samuel's study was better when the word was predictable, indicating the importance of context. He concluded that the contextual information did not have a direct effect on bottom-up processing and that it influences the listeners' expectations in a top-down fashion, but these expectations then needed to be confirmed with reference to the sound that was actually presented [25].

1.2 Advantages and Limitations of Two Schemes

Top-down and bottom-up are strategies of information processing, which have been used in a variety of studies, including psychology, management, and software design.

In a top-down approach, an overview of the system is first formulated, specifying but not detailing any first-level subsystems. Each subsystem is then refined in greater detail, sometimes in many additional subsystem levels, until the entire specification is reduced to primitive elements. A top-down model is often specified with the assistance of "black boxes," which make it easier to manipulate. However, black boxes may fail to elucidate elementary mechanisms or be detailed enough to realistically validate the model.

In a bottom-up approach, the individual base elements of the system are first specified in great detail. These elements are then linked together to form larger subsystems, which then in turn are linked, sometimes at many levels, until a complete top-level system is formed. This strategy often resembles a "seed" model, whereby the

beginnings are small but eventually grow in complexity and completeness. However, this procedure may result in a tangle of elements and subsystems due to isolated development and may be subject to local optimization as opposed to meeting a global purpose.

We can interpret some pattern recognition techniques from a perception perspective. For instance, the well-known hidden Markov model (HMM)-based automatic speech recognition system can be regarded as a top-down framework, which explicitly puts all the available knowledge, practical constraints, and conceptual expectations into a huge network [44] that cannot be easily modified over time. It is an expectation-driven framework and some expectations are imposed on the system's input. It is also a "black-box," which means that when the system's behavior is different from our expectation, it is hard to diagnose and figure out a solution to the problem. Due to the scarcity of data, it is always impossible to completely and accurately model the inherent uncertainties in the data during the system development stage. Therefore, the top-down framework tends to be fragile, with mismatches always occurring, and it cannot effectively handle many of the uncertainties and complexities in real-world situations.

The *eigenface* method [101] for face detection and recognition is another example of top-down processing in computer vision. The system analyzes a human face based on patterns known as eigenfaces, which are represented through a series of high-dimensional vectors derived from statistical analysis of many face images. It tries to model a face image as a whole and each face image is represented by a vector in a high-dimensional space and then projected to an eigenface space spanned by the eigenvectors of a set of face images. However, in real-world conditions of light variance, partial occlusion, and image deformation, the face detection performance quickly degrades.

We can conclude from the examples above that a top-down processing framework is

a "black-box" insufficient for dealing with the kinds of variations occurring in practical pattern recognition systems, while for the bottom-up or stimulus-driven processing, its failure to adequately consider top-down processing is its greatest limitation. The relative importance of top-down and bottom-up processing depends on various factors. For instance, visual perception can be dependent mostly on bottom-up processing when the viewing conditions are good, but will need to involve top-down processing as the viewing conditions deteriorate because of lack of stimulus clarity.

1.3 Detection-based Pattern Recognition Framework

Inspired by the studies of modern cognitive psychology and some practical pattern recognition applications, it is natural to propose a detection-based pattern recognition framework, within which the primitive features are first detected and the knowledge hierarchy is constructed level by level; then a variety of heterogeneous information sources are combined together and the high level context is incorporated as additional information at certain stages.

A detection-based framework is a "divide-and-conquer" design paradigm for pattern recognition problems, which will decompose a conceptually difficult problem into many elementary subproblems that can be handled directly and reliably. Some information fusion strategies will be employed to integrate the evidence from a lower level to form the evidence at a higher level. Such a fusion procedure continues until reaching the top level. Generally, a detection-based framework has many advantages: (1) more flexibility in both detector design and fusion strategies, as these two parts can be optimized separately; (2) parallel and distributed computational components in primitive feature detection. In such a component-based framework, any primitive component can be replaced by a new one while other components remain unchanged; (3) incremental information integration; (4) high level context information as additional information sources, which can be combined with bottom-up processing at any

stage.

1.3.1 Detector Design and Knowledge Hierarchy

The first component of a detection-based framework is a collection of primitive feature detectors and the task-specific knowledge hierarchy. In principle, detector design is similar to the matched filter design in communication systems for artificial signals. However, it is more difficult to design detectors for natural signals like speech and video.

Statistical decision and detection theory provides the theoretical foundation for optimal detector design, detector comparison, and hypothesis verification. We can design detectors based on different criteria, e.g., Bayes criterion, Min-Max criterion, and Neyman-Pearson criterion [52]. Each individual detector can be optimally designed to detect a particular event reliably. In addition, the coverage and completeness of primitive features need to be considered thoroughly.

Two kinds of primitive feature detectors were investigated in this dissertation: knowledge-guided signal processing methods and data-driven model-based methods. These approaches will be used in different scenarios. When task-specific knowledge is explicitly presented, knowledge-guided methods will be preferred. This is because signal processing methods have the advantage of requiring only a small amount of training data to achieve good detection performance, with the help of domain-specific knowledge. For instance, phonetic and phonological knowledge were used in landmark detection of speech signals. In some cases, data-driven model-based methods will outperform signal processing methods.

For data-driven methods, unsupervised, semi-supervised, and supervised learning methods are employed regarding the amount of labeled training data. A multi-modal spectral clustering algorithm is proposed and investigated, which is different from

hierarchical clustering algorithms because only a pair-wise adjacency matrix is required. When the amount of the labeled training data is relatively small, a probabilistic model-based semi-supervised learning approach is proposed to utilize the large amount of unlabeled training data. Both theoretical analysis and experimental studies were conducted.

Designing detectors that are robust to imperfect measurements and noisy conditions is always a challenging task. It is well-known that a point on receiver operating characteristics (ROC) or precision-recall (P-R) curves represents a potential operating point. A novel algorithm is proposed in this dissertation to optimize the area under the ROC and P-R curve. By maximizing the area under curve (AUC), robust detectors that have the best performance for all practical operating points can be obtained.

1.3.2 Information Integration

Within the knowledge hierarchy of a detection-based framework, the higher level knowledge can be obtained by combining its lower level knowledge sources and level-specific information. For instance, the knowledge about phonemes could be obtained from the knowledge of distinctive features and phonological rules. Knowledge integration will be carried out at each level of a knowledge hierarchy, such as signal (sensor), feature, and decision level.

At the feature level, feature concatenation is a simple combination scheme, while at the decision level, some intuitive knowledge integration strategies are usually used, such as majority voting, weighted majority voting, and arithmetic average. Knowledge integration can be conducted in either a parallel manner, such as a logistical combination of classifiers, or a sequential (cascade) fashion, such as a decision tree. The basic principle is to gradually remove false alarms while keeping the true positive probability as high as possible for meaningful detection by adding more evidence and

constraints to the fusion procedure.

Knowledge integration is similar to ensemble classifiers in some sense, such as bagging [7], boosting [93], stacking, and so on. Generally, ensemble classifiers work on the same feature set through different kinds of re-sampling and weighting techniques for each classifier, while evidence fusion focuses on integrating heterogeneous knowledge sources from different detectors.

Feature concatenation and model combination schemes were first discussed and then the maximum entropy (MaxEnt) model was investigated, which is a log-linear model for combining heterogeneous sources. Finally, a novel discriminative fusion strategy, regularized maximum figure-of-merit (rMFoM) approach was proposed, which is a discriminative fusion strategy working with any discriminant function, such as linear discriminant functions (LDF) used in both MaxEnt models and support vector machines (SVMs).

1.4 Summary and Reading Guide

The objective of this dissertation is to present a detection-based pattern recognition framework and then to demonstrate its applications in automatic speech recognition and broadcast news video story segmentation.

Inspired by the studies of human vision and auditory perception, a "divide-and-conquer" strategy was used with primitive feature detection and decision combination. In contrast to the conventional top-down frameworks, a detection-based framework first detects a collection of candidate hypotheses and events. Then, knowledge-guided evidence fusion and hypothesis verification strategies are used to prune false alarms. Evidence fusion is conducted in an incremental manner, which means that as more evidence and constraints are available, the hypotheses will become sharper and more focused until a consistent decision can be made.

This dissertation is organized as follows:

In Chapter 2, basic principles for detector design and hypothesis verification based on statistical detection and decision theory are presented. Two novel detection approaches were proposed in this dissertation: a supervised robust detector design algorithm and a semi-supervised model-based detector design algorithm. Experimental results on various datasets clearly demonstrate the effectiveness of the proposed methods.

In Chapter 3, commonly used information integration methods such as feature concatenation, model combination, and the maximum entropy (MaxEnt) model are discussed first for combining heterogeneous knowledge sources and then a regularized maximum-figure-of-merit (rMFoM) learning algorithm is proposed for information fusion.

In Chapter 4, we present our study on detection-based automatic speech recognition. First, several low level detectors are described, such as landmark detection, manner and place of articulation detection, and then the MaxEnt model is used to combine the low level attributes into phoneme level for phoneme classification. In addition, these low level features have been incorporated into both digit recognition and large vocabulary continuous speech recognition (LVCSR) systems by hard and soft decisions, respectively. Finally, a comparative study was conducted to compare four commonly used combination schemes in LVCSR systems.

In Chapter 5, we demonstrate a detection-based broadcast news video story segmentation system. First, multi-modality features were detected by different methods, such as anchor shot detection using unsupervised multi-modal spectral clustering and semantic concepts detection by supervised robust detectors. Then, the regularized maximum figure-of-merit (rMFoM) method was used to combine these multi-modality features to predict story boundaries.

Chapter 6 concludes the studies of this dissertation. The contributions from this

dissertation are highlighted and summarized. Meanwhile, some future research directions are also briefly discussed.

CHAPTER II

DETECTOR DESIGN

A detection-based pattern recognition framework consists of several components: a collection of primitive feature detectors, construction of the knowledge hierarchy, and some knowledge integration schemes. It decomposes the total variability, uncertainties, and complexity of a large system into local ones. Therefore, it is very flexible in optimal detector design and fusion strategy optimization.

Although the detector design for low level primitive features is application-dependent, some common criteria and techniques are shared by a variety of real-world applications. In the following sections, the theoretical foundation of detector design will be briefly presented at first. Then, some detector design criteria and techniques will be discussed in detail, including signal processing methods and data-driven model-based methods.

2.1 Statistical Decision and Detection Theory

Statistical decision and detection theory provide the theoretical foundation for optimal detector design, detector comparison, and hypothesis verification.

Statistical decision theory [10] [103] is all about making the best decision and inference under uncertainty. It is a set of concepts applied to statistical hypothesis testing, signal detection theory, and many others (e.g., parameter estimation). Utility functions are generally used to express our preference for consequences and the inductive use of probability is employed to express our knowledge about uncertainties. A decision function is used to incorporate the cost of different types of errors and prior information for each class into a single performance measure. The accuracy of a decision is usually measured by a loss function. The risk is a function of the

decision rule, and is defined to be the expectation of a loss with respect to the joint distribution.

Detection is essentially a statistical hypothesis testing problem. Detection theory has been studied intensively in statistical hypothesis testing [81] and statistical signal processing [52]. A null hypothesis H_0 states that a specific "feature" is absent, while the alternative hypothesis H_1 states that it is present. Figure 1 shows how to perform a one-sided hypothesis test concerning a testing statistics y. It also illustrated the conditional distributions of testing statistics y given H_0 and H_1, respectively. By changing the threshold γ, a detector can work at different operating points. Its performance can be indicated as P_F and P_D, which are called the size and power of a test, respectively.

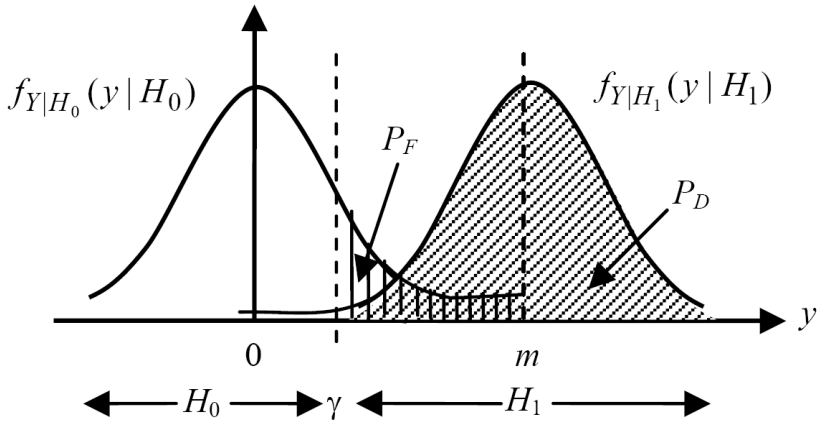

Figure 1: Statistical hypothesis testing.

Table 1 and Table 2 show the statistical hypothesis testing and statistical signal detection models. Their relationship and differences can be clearly observed.

Table 1: Statistical hypothesis testing model.

Decision	Reality	
	H_0 true ($\mu_1 - \mu_2 = 0$)	H_0 false ($\mu_1 - \mu_2 \neq 0$)
Do not reject H_0	correct decision	type II error, P(II) = β
Reject H_0	type I error, P(I) = α	correct decision

Table 2: Statistical signal detection model.

Decision	Reality	
	Noise only (N)	Noise + Signal (NS)
Noise	correct rejection	Miss
Signal	False Alarm	Correct Detection

Table 3: Neyman-Pearson criterion.

	Statistics		Detection Theory	
Prob.	Name	Notation	Name	Notation
$P(R\|H_0)$	size	α	false-alarm prob.	P_F
$P(R\|H_1)$	power	1 - β	detection prob.	P_D

All the relevant quantities regarding Neyman-Pearson theory are summarized in Table 3, where α and β are called type-I error and type-II error, respectively. P_F and P_D are related to each other through the rejection region R of the null hypothesis. Therefore, P_F and P_D represent the two degrees of freedom in a binary hypothesis test. They represent the fundamental trade-off in hypothesis testing and detection theory. Under the Neyman-Pearson criterion, a significance level α (the size of the test) is specified on the false alarm probability (type-I error), and one seeks a detector that satisfies this constraint while minimizing the miss probability (type-II error), or equivalently maximizing the detection probability (power). The Neyman-Pearson lemma specifies necessary and sufficient conditions for the most powerful test of size α, provided the distributions under the two hypotheses are known, or (in special cases) the likelihood ratio is a monotonic function of an unknown parameter. The Neyman-Pearson criterion offers an alternative to the Bayesian framework when different types of error have different consequences or the priori probabilities are unknown. Neyman-Pearson theory states that for a simple point hypothesis, the likelihood ratio test is the most powerful test. It means that at a given type-I error α, this likelihood ratio detector can achieve the maximal power, or equivalently, minimal type-II error, β. This is the theoretical foundation for verification and likelihood ratio test.

2.2 Detector Performance Metrics

The performance of a detector can be measured both locally and globally. For a given operating point γ, we could use error rate, recall, precision, or other cost functions to measure its local performance. A receiver operating characteristics (ROC) graph [26] is commonly used to display the relationship between P_D and P_F for all possible operating points and therefore shows the global performance of a detector. For many detection problems, an single error rate or classification accuracy cannot indicate the system performance correctly, especially for imbalanced data. In this type of asymmetric situation, other performance metrics such as recall, precision, and F1 measure are more appropriate.

The x-axis of a ROC graph is the false positive probability P_F and the y-axis is the true-positive probability P_D. A ROC graph is a useful tool for detector performance analysis and visualization. ROC analysis provides tools to select optimal models and discard suboptimal ones independently from the cost context or the class distribution. It is also related in a direct and natural way to cost/benefit analysis of diagnostic decision making. A proper detector always has its ROC curve on the left-top part of the graph, which means that $P_D \geq P_F$ for every operating point.

Figure 2 shows several different operating points on the ROC graph of a detector. For instance, the point at the upper left corner indicates a perfect detector. The one at the top shows a low missing error and another one at the middle left shows a low false alarm error. Similarly, a recall-precision graph could be used to indicate the performance of both detection and ranking. We will design detectors that work at different operating points for different purposes. For instance, candidate detection needs to have a maximal recall [68].

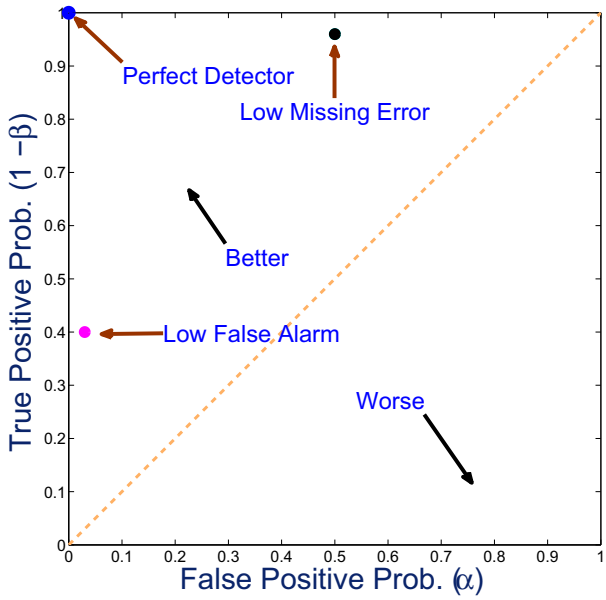

Figure 2: A ROC graph of a probabilistic detector.

2.3 Detector Design Principles, Criteria, and Techniques

At different levels of a knowledge hierarchy, different requirements are imposed on detector design. At the candidate level, high detection rate is expected to avoid candidate missing, whereas at the decision level, the detector design is expected to match the performance evaluation metrics. The different detectors correspond to different points in the ROC or recall-precision graph and they are working under different operating points. Therefore, they provide supplemental information for further performance improvement. For instance, Figure 3 shows four detectors for broadcast news video story segmentation; some of them have better coverage, while some have better precision.

All of the existing binary classification techniques, such as artificial neural networks (ANNs) [5], support vector machines (SVMs) [8], and HMMs [87] could be used at different levels of a knowledge hierarchy. For example, we need word detectors, sub-word detectors, and attribute detectors in ASR, and event detectors from image, audio, text, and so on for video story segmentation.

Most binary classification algorithms are designed to minimize the probability of

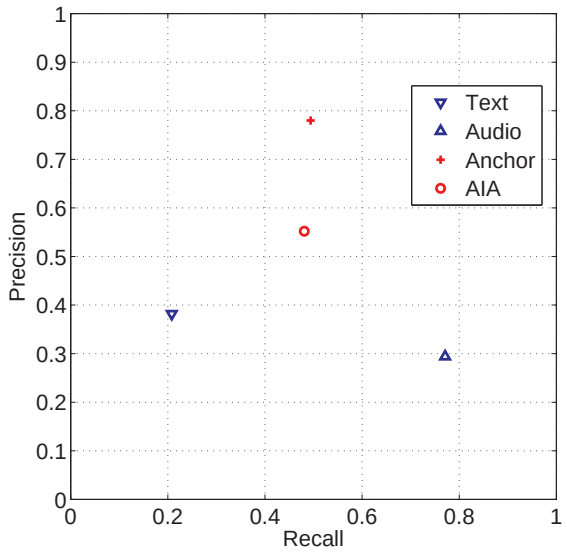

Figure 3: Four detectors in recall-precision space.

errors. In some situations, the cost of the two types of errors is asymmetric and the distribution of the samples from two classes is imbalanced, such as fraud detection, spam filtering, and novelty detection, just to name several examples. Therefore, some of the commonly used binary classification algorithms are not suitable for detection. When the class prior information and the misclassification cost are unknown, the Neyman-Pearson criterion provides a way to design optimal detectors. It first specifies a significance level, and then finds the most powerful detector [81]. With the Neyman-Pearson criterion, a very low missing probability (or a very high detection probability) is specified, while keeping the false alarm probability as small as possible. When the misclassification cost is known, some cost-sensitive learning schemes can be used to optimize the expected loss on the training data [24]. In addition, when both the prior and misclassification cost information are available, Bayesian decision theory can be used to find a detector with minimal expected cost [10] [23].

2.4 Construction of Knowledge Hierarchy

The first step of a detection-based framework is to build the infrastructure of a task-specific knowledge hierarchy, which represents the domain-specific knowledge sources

at different levels, different resolutions, and their dependency structure. Such a knowledge hierarchy would be shared among many different applications. For example, the meta knowledge detected from speech signals can be used in speech recognition, speaker identification, and language identification. As our understanding of speech becomes more mature, the knowledge hierarchy of speech will cover more intrinsic properties of speech.

For any level in a knowledge hierarchy, there is knowledge from its lower levels and its level-specific information. Fusion strategies could be employed to combine the lower level information and level-specific information to construct the knowledge hierarchy. For instance, landmarks, manner and place of articulation, and phonological rules could be used to identify phonemes. As this procedure repeats itself, the task-specific knowledge hierarchy could be constructed level-by-level in a bottom-up manner.

2.5 Signal Processing Methods

Matched filters are the optimal linear detectors that maximize the signal-to-noise ratio (SNR) for artificial signals used in communication systems. Detector design for natural signals (e.g., speech and video) is much more difficult.

With the help of domain-specific knowledge, signal processing methods can be invented to detect low level attributes efficiently and effectively. For instance, landmark detection and voice activity detection (VAD) in speech signals are conducted using signal processing methods. Conversely, when task-specific knowledge is explicitly presented, knowledge-guided methods will be preferred. For instance, phonetic and phonological knowledge were used in landmark detection of speech signals. One of the advantages of signal processing methods is that only a small amount of training data is needed to achieve good detection performance with the help of domain-specific knowledge.

2.6 Supervised Robust Detector Design

As stated in previous sections, if we do not have the class prior information and the cost of misclassification, the Neyman-Pearson criterion is one way to design optimal detectors. However, the detector designed under the Neyman-Pearson criterion is optimal only at a single operating point. It has no performance guarantee on other operating points. Sometimes, it is desirable to have robust detectors that have good performance on other operating points as well. Such an optimal detector should have maximal power at size α in the sense of the Neyman-Pearson criterion. For general cases, it is not easy to find such a uniformly most powerful detector.

Table 4: Contingency table.

		Test	
		positive	negative
Reference	positive	True Positive (TP)	False Negative (FN)
	negative	False Positive (FP)	True Negative (TN)

Almost all the metrics used in detection performance evaluation can be computed from the contingency table shown in Table 4, such as error rate (Err), recall, precision, and F1 measure, which are computed as follows:

$$\text{Err} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \text{ and} \tag{3}$$

$$\text{F1} = \frac{2\text{TP}}{\text{FP} + \text{FN} + 2\text{TP}}. \tag{4}$$

Based on an analytic approximation of the counts in the table, these performance metrics can be optimized in a discriminative manner [31]. As in the Neyman-Pearson criterion, these metrics are used to measure the detector's performance at a single operating point.

Previous studies showed that by optimizing the area under a ROC curve, a robust

classifier can be obtained [26]. Studies also show that the area under the ROC curve is equivalent to the Wilcoxon-Mann-Whitney (WMW) rank statistics [30].

Similarly, the area under a precision-recall curve is the performance metric used in many information retrieval systems, such as automatic image annotation (AIA) systems, which return a ranked list of relevant images according to a confidence score from a collection of images for each semantic concept. An average precision (AP) is the average of the precision scores at the rank locations of each relevant image, which emphasizes the returning of more relevant images earlier. It can be considered as the measure of the area under the precision-recall curve. There are several analytic approximations for these two metrics. For instance, some discriminative learning algorithms have been investigated to optimize the recall, precision, accuracy, and F1 measure on the training set [31]. In addition, approaches such as an ensemble learning framework and support vector methods were proposed to optimize the area under the ROC curve [30] [64]. However, when working on large training data set, optimizing AP and the area under the ROC curve require heavy computations at each iteration. So in this section, a numerical approximation algorithm is proposed to design discriminant function based robust detectors.

In this study, we proposed an efficient gradient computation approach for robust detector design such that the model parameters can be estimated by directly optimizing the performance metric AP on very large training datasets.

2.6.1 Non-linear Performance Metrics

The above-mentioned performance metrics such as recall, precision, and F1 measure are non-linear performance metrics, which cannot be decomposed into a sum of the loss over individual instances. These performance metrics are obtained from an unordered score set and can be computed directly from the four items in a contingency table. However, other non-linear performance metrics, such as the area

under the ROC curve (AUC-ROC) [105] and the area under the precision-recall curve (AUC-PR) [105] cannot be computed from a contingency table, because they require predicting a ranking function instead of a classification function. Studies showed that these two performance metrics are inconsistent in some cases, i.e., optimizing one of them cannot guarantee an optimal value for the other [20]. By its definition, we know that average precision (AP) used in some detection evaluation is equivalent to AUC-PR, which can be expressed as follows:

$$\text{AUC-PR} \quad = \quad \frac{1}{M}\sum_{i=1}^{M}\frac{i}{\sum_{j=1}^{N} I(s_i^+, s_j^-) + i}, \tag{5}$$

where $I(s_i^+, s_j^-)$ is an indicator function that is 1 if $s_i^+ < s_j^-$ and zero otherwise, s_i^+ is the i^{th} highest positive score and s_j^- is the j^{th} negative score. M and N are the number of positive and negative samples, respectively. Smooth differentiable functions can be used to approximate the discrete ranking count indicator function analytically, such as sigmoid functions [30]. By this way, we can compute the gradient of AUC-PR by a sum of the gradient of all $M * N$ pairs-wise gradient from this analytic form of the indicator function.

In practical systems, representing all $M * N$ pairs would be rather inefficient when $M * N$ is a huge number as in a semantic concept detection task. So in this study, we present an efficient numerical approximation approach that requires only $M + N$ gradient computation in every iteration of gradient descent optimization. The proposed method and the MFoM method [31] [30] differ in two aspects: (1) In the MFoM method, the gradient is calculated from an analytically smoothed indicator function, while in the proposed approach, the gradient is calculated by small perturbation in the ranking score space; (2) In the MFoM method, the gradient calculation consists of $M * N$ pair-wise gradients from each indicator function; while in our approach, the gradient calculation consists of $M + N$ point-wise gradients from each observation instance. In most cases, the performance metrics optimization will be more efficient. It should be noted that the individual gradient calculation in our approach is more

computationally expensive than that in the MFoM method.

2.6.2 Discriminant Functions and Decision Scores

The proposed robust detector is based on the notation of discriminant functions. The discriminative robust detector for each event consists of two discriminant functions: one for the positive instances and the other one for the negative instances. They are represented as $f_{\texttt{pos}}(x)$ and $f_{\texttt{neg}}(x)$, respectively.

For each observation sample x, its uncalibrated ranking score is computed as follows:

$$s(x;\theta) \quad = \quad f_{\texttt{pos}}(x;\theta_{\texttt{pos}}) - f_{\texttt{neg}}(x;\theta_{\texttt{neg}}), \tag{6}$$

where $\theta = (\theta_{\texttt{pos}}, \theta_{\texttt{neg}})$ represents the model parameters. There are many choices for discriminant functions, such as linear discriminant functions (LDFs), log-likelihood of class conditional probability density functions (e.g., Gaussian mixture models and hidden Markov models). $f_{\texttt{neg}}(x;\theta_{\texttt{neg}})$ could be a single discriminant function when focusing on single events. It could also be a combination of multiple discriminant functions as well when dealing with multiple event detection simultaneously. For instance, it could be a geometrical mean of the discriminant functions of several competing event [31].

2.6.3 Gradient Calculation

To optimize AUC-PR given a collection of training instances, its derivatives with respect to each score will be numerically approximated. The AUC-PR is a function of all the ranking scores from both positive and negative samples, AUC-PR = $f(s_1, s_2, \cdots, s_{M+N})$. It is a function in the score space and also a function of $\theta_{\texttt{pos}}$ and $\theta_{\texttt{neg}}$. The AUC-PR gradient at each score can be computed from the change of AUC-PR by a small perturbation in score from s_i to $s_i + \delta$, $\nabla\text{AUC-PR}(s_i) = \Delta\text{AUC-PR}/\delta$. To get a more reliable and smooth gradient estimation for each training instance, an

average of the gradient from different perturbation step sizes are used.

The numerical approximation of the gradient at each score s_i is expressed as follows:

$$\nabla\text{AUC-PR}(s_i) \approx \frac{1}{2L}\sum_{\substack{l=-L \\ l\neq 0}}^{L} \nabla\text{AUC-PR}(s_i + l * \delta). \tag{7}$$

Here δ is a step size decided by the spread of the ranking scores. L is used to control the smoothness of the obtained gradient curve.

2.6.4 Parameter Update

We aim to maximize the AUC-PR by updating the parameters of both positive and negative discriminant functions. Either the batch gradient descent or the generalized probabilistic descent (GPD) algorithm can be used to optimize the model parameters on the training set [31]. For each training instance x, the gradient for the input data and current model parameters is used to update model parameters. By using the chain rule of the derivative, AP can be maximized by updating the parameters $\theta_{\texttt{pos}}$ and $\theta_{\texttt{neg}}$ of the discriminant functions,

$$\theta_{t+1} = \theta_t + \gamma \sum_{i=1}^{M+N} \frac{\partial s_i}{\partial \theta} \nabla\text{AUC-PR}(s_i). \tag{8}$$

To speed-up the parameter update procedure, we can use a larger step size γ. Another way is to perform instance sampling at each iteration based on the obtained score list given to current model parameters. Actually, this is necessary when dealing with imbalanced training data. Generally, only a small number of positive instances are available. By instance re-sampling, we can keep positive and negative instance balanced and none of them will dominate the gradient calculation. After the ranking scores for all the instances are computed based on the current model parameters, both positive and negative scores are sorted. Then a uniform sampling is conducted on these two sorted list independently, which is equivalent to a sampling from the distribution of both positive and negative scores.

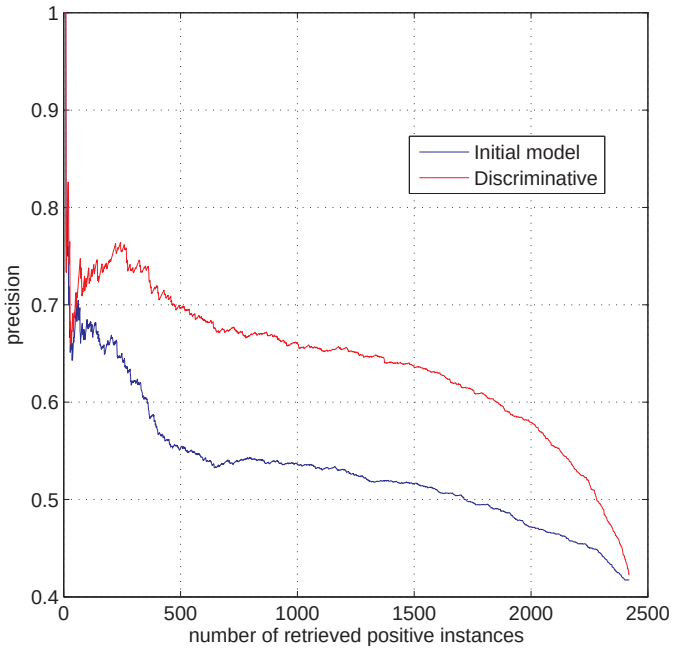

Figure 4: Robust detector design results.

Figure 4 shows the result of a robust detector design scheme. It's clear that the area under the precision-recall curve have been increased. The discriminant functions are Gaussian mixture models and the EM algorithm is used to initialize the model parameters. The AP has been improved from 0.53 to 0.85 on the training set and from 0.53 to 0.75 on the testing set.

2.7 Semi-supervised Model-based Detector Design

In some situations, even for a single attribute or event, there is not enough data for model training, or it is too expensive to collect sufficient labeled data. So in this section, a semi-supervised statistical model-based learning scheme is proposed to leverage the large amount of unlabeled data and limited labeled data.

The target value y associated with each observation vector x is discrete and finite, $y \in \{1, 2, \cdots, K\}$. In supervised learning, the label y for each x in the training set is provided. This supervision information can be represented by a label distribution $\phi_x(y)$ for each x. The label distribution is used to represent the class membership

for each observation x and it is a probability mass function. With the notion of the label distribution, full supervision (e.g., $\phi_x(y) = [1, 0]$), partial supervision (e.g., $\phi_x(y) = [0.7, 0.3]$), and no supervision (e.g., $\phi_x(y) = [1/2, 1/2]$) can be expressed in a unified manner. We assume that all pairs (x, y) are independent and identically distributed (i.i.d.) samples from an unknown distribution $p(x, y; \theta)$. Here, θ are the parameters of a generative model. For detection tasks, it includes the parameters of both positive and negative models.

We are aiming to incorporate unlabeled data into the modeling process; so a labeling strategy is necessary. Given the current model parameters θ, a posterior probability distribution $P(y|x; \theta)$ can be estimated using Bayes rule for each observation sample x. A labeling strategy is used to generate a label distribution from the posterior distribution. There are two kinds of labeling strategies: soft decision and hard decision. In the soft decision strategy, we always take $\phi_x(y) = P(y|x; \theta)$, which means that the label distribution is identical to the posterior distribution for each sample, while in the hard decision strategy, the sample will be labeled as the class with the highest posterior probability. For original labeled data, their label distribution will be fixed during the learning process, while for unlabeled samples, their label distributions $\phi_x(y)$ will be updated at each iteration.

In semi-supervised learning settings [13], only a small part of the observation samples has full supervision and most of them are unlabeled. The whole training data X can be represented as the union of two subsets X_l and X_u, where $X_l = \{x_i, i = 1, \cdots, N\}$ is the labeled data and $X_u = \{x_i, i = N+1, \cdots, N+M\}$, $M \gg N$ is the unlabeled data.

With the assumed joint distribution $p(x, y; \theta)$, the log-likelihood for single observation x is defined as follows:

$$L_x(\theta) \quad := \quad \log \sum_{y=1}^{K} p(x, y; \theta). \tag{9}$$

The log-likelihood over the whole training set X is defined as follows and does not

depend on the label distribution $\phi_x(y)$:

$$\mathcal{L}(\theta) \quad = \quad \sum_{x \in X} L_x(\theta). \tag{10}$$

The following function is defined for semi-supervised learning for single observation x:

$$F_x(\phi_x(y); \theta) \quad := \quad L_x(\theta) - \lambda KL(\phi_x(y) \| P(y|x; \theta)). \tag{11}$$

Here, $KL(\phi_x(y) \| P(y|x; \theta))$ is the Kullback-Leibler divergence [17] between the label distribution $\phi_x(y)$ and the posterior distribution $P(y|x; \theta)$ given the current model parameters. With the properties of KL-divergence, we have the following inequality:

$$F_x(\phi_x(y); \theta) \quad \leq \quad L_x(\theta). \tag{12}$$

The equality holds if and only if

$$KL(\phi_x(y) \| P(y|x; \theta)) = 0 \quad \Leftrightarrow \quad \phi_x(y) = P(y|x; \theta). \tag{13}$$

The proposed objective function of the whole training set X is the following:

$$Q(\phi_x(y), \theta) \quad := \quad \sum_{x \in X_l} F(\phi_x(y), \theta) + C \sum_{x \in X_u} F(\phi_x(y), \theta). \tag{14}$$

It means that this function is a lower bound of the likelihood function of the joint distribution and it is a combination of the log-likelihood and the KL-divergence. With this objective function, we aim to maximize the log-likelihood of each sample (better data fitting), meanwhile keeping the divergence between the label distribution and the predicted label distribution as small as possible (better label agreement). λ is a scaling factor to make the two parts comparable and C is used to control the contribution from labeled data.

In the following, a detailed discussion and theoretical justification of this objective function are presented. The batch gradient descent algorithm or the stochastic gradient descent algorithm could be used to estimate the parameters θ and the expectation maximization (EM) will be used in the following sections.

Case 1: In the case of the fully unlabeled data and soft decision labeling strategy, the proposed objective function is equivalent to the maximum likelihood (ML) criterion. The EM algorithm is a method for estimating ML parameters of a model with missing or latent variables [22] and it is a maximization-maximization procedure of the proposed objective function $Q(\phi_x(y), \theta)$ with regard to $\phi_x(y)$ and θ, respectively.

The E-step is actually a relabeling procedure and the aim is to find an optimal $\phi_x^{(t)}(y)$ given that $\theta^{(t-1)}$ is fixed.

$$\begin{aligned} \phi_x^{(t)}(y) &= \arg\max_{\phi_x(y)} F(\phi_x(y), \theta^{(t-1)}) \\ &= \arg\min_{\phi_x(y)} KL(\phi_x(y) \| P(y|x, \theta^{(t-1)})) \\ &= P(y|x, \theta^{(t-1)}) \end{aligned} \tag{15}$$

So, the following inequality always holds by this relabeling strategy for any observation x from both X_l and X_u:

$$F(\phi_x^{(t)}(y), \theta^{(t-1)}) \geq F(\phi_x^{(t-1)}(y), \theta^{(t-1)}) \quad (\forall\, x \in X). \tag{16}$$

Moreover, if we take $\phi_x^{(t)}(y) = P(y|x, \theta^{(t-1)})$, we get the following:

$$L_x(\theta^{(t-1)}) = F(\phi_x^{(t)}(y), \theta^{(t-1)}). \tag{17}$$

The M-step is a re-estimation procedure to find the optimal parameter $\theta^{(t)}$ over the whole training data X given that the label distribution $\phi_x^{(t)}(y)$ of each observation x is fixed as shown:

$$\begin{aligned} \theta^{(t)} &= \arg\max_{\theta} Q(\phi_x^{(t)}(y), \theta) \\ &= \arg\max_{\theta} \sum_{x \in X_l} F(\phi_x^{(t)}(y), \theta) + C \sum_{x \in X_u} F(\phi_x^{(t)}(y), \theta). \end{aligned} \tag{18}$$

Whatever the label distribution $\phi_x^{(t)}$ is, when θ is chosen according to Equation (18), the following inequality always holds:

$$Q(\phi_x^{(t)}(y), \theta^{(t)}) \geq Q(\phi_x^{(t)}(y), \theta^{(t-1)}). \tag{19}$$

From Equations 16, 17, and 18, it is easy to see that

$$\begin{aligned} \mathcal{L}(\theta^{(t)}) &= Q(\phi_x^{(t+1)}(y), \theta^{(t)}) \geq Q(\phi_x^{(t)}(y), \theta^{(t)}) \\ &\geq Q(\phi_x^{(t)}(y), \theta^{(t-1)}) = \mathcal{L}(\theta^{(t-1)}). \end{aligned} \tag{20}$$

This means that the EM algorithm will increase the proposed objective function iteratively until it converges to a local maximizer. This is the well-known monotonicity property of the EM algorithm.

Case 2: Now for the semi-supervised learning settings, some constraints are imposed on the relabeling strategy, which is as follows:

$$\phi_x^{(t)}(y) = \begin{cases} \phi_x^{(0)}(y) & x \in X_l \\ P(y|x, \theta^{(t-1)}) & x \in X_u. \end{cases} \tag{21}$$

This means that for the original labeled data, their label distributions will remain unchanged. So this constrained relabeling strategy can guarantee that inequality (16) still holds for both $x \in X_l$ and $x \in X_u$. However, the equality (17) will never hold for $x \in X_l$.

The re-estimation step is the same as that in Case 1. Overall, we have the following inequalities:

$$\begin{aligned} Q(\phi_x^{(t+1)}(y), \theta^{(t)}) &\geq Q(\phi_x^{(t)}(y), \theta^{(t)}) \\ &\geq Q(\phi_x^{(t)}(y), \theta^{(t-1)}). \end{aligned} \tag{22}$$

Case 3: Now let's change the soft-decision labeling strategy in Case 2 for $x \in X_u$ to the hard-decision labeling strategy. It is equivalent to imposing some constraints on the label distribution $\phi_x(y)$ such that the entropy of the label distribution $\phi_x(y)$ of the newly labeled data must be zero, $H(\phi_x^{(t)}(y)) = 0$. So an unlabeled data will be definitely assigned to one class by "hard decision" instead of "soft assignment" used

in Case 2. Now we can rearrange the objective function a little bit as follows:

$$\begin{aligned} Q(\theta, \phi_x(y)) &= \sum_{x \in X} F(\phi_x(y), \theta) \\ &= \sum_{x \in X} (L_x(\theta^{(t-1)}) - D(\phi_x(y) \| P(y|x, \theta))) \\ &= \sum_{x \in X} (L_x(\theta^{(t-1)}) - H(\phi_x(y) \| P(y|x, \theta)) + \underbrace{H(\phi_x(y))}_{=0}) \\ &\triangleq \sum_{x \in X} G(\phi_x(y), \theta). \end{aligned} \tag{23}$$

Here, $H(\phi_x(y) \| P(y|x, \theta))$ is the cross-entropy defined as follows in information theory [17]:

$$H(\phi_x(y) \| P(y|x, \theta)) = D(\phi_x(y) \| P(y|x, \theta)) + H(\phi_x(y)). \tag{24}$$

The E-step (relabeling) is to find an optimal $\phi_x^{(t)}(y)$ with $H(\phi_x^{(t)}(y)) = 0$, given that $\theta^{(t-1)}$ is fixed:

$$\begin{aligned} \phi_x^{(t)}(y) &= \arg\max_{\phi_x(y), H(\phi_x(y))=0} G(\phi_x(y), \theta^{(t-1)}) \\ &= \arg\max_{\phi_x(y), H(\phi_x(y))=0} (L_x(\theta^{(t-1)}) - H(\phi_x(y) \| P(y|x, \theta^{(t-1)}))) \\ &= \arg\min_{\phi_x(y), H(\phi_x(y))=0} H(\phi_x(y) \| P(y|x, \theta^{(t-1)})). \end{aligned} \tag{25}$$

Based on the property of cross-entropy [17] and our constraints, the $\phi_x^{(t)}(y)$ can be obtained by assigning the observation to the class with the highest predictive probability $P(y|x, \theta^{(t-1)})$.

By using such a re-labeling strategy, the following inequality holds:

$$G(\phi_x^{(t)}(y), \theta^{(t-1)}) \geq G(\phi_x^{(t-1)}(y), \theta^{(t-1)}) \quad (\forall\, x \in X). \tag{26}$$

The M-step (re-estimation) is to find the optimal $\theta^{(t)}$ given that $\phi_x^{(t)}(y)$ is fixed:

$$\begin{aligned} \theta^{(t)} &= \arg\max_{\theta} \sum_{x \in X} G(\phi_x^{(t)}(y), \theta) \\ &= \arg\max_{\theta} \sum_{x \in X} \sum_{k=1}^{K} \phi_x^{(t)}(y = k) \log p(x, y|\theta). \end{aligned} \tag{27}$$

Then we have

$$\sum_{x\in X} G(\phi_x^{(t)}(y), \theta^{(t)}) \geq \sum_{x\in X} G(\phi_x^{(t)}(y), \theta^{(t-1)}). \tag{28}$$

From the above inequalities, it's easy to see that

$$\begin{aligned} \sum_{x\in X} G(\phi_x^{(t+1)}(y), \theta^{(t)}) &\geq \sum_{x\in X} G(\phi_x^{(t)}(y), \theta^{(t)}) \\ &\geq \sum_{x\in X} G(\phi_x^{(t)}(y), \theta^{(t-1)}). \end{aligned} \tag{29}$$

Case 4: Until now, all the training data X are used in the re-estimation step in a batch mode. It is desirable to have the unlabeled data labeled in an incremental way. In the following, an incremental self-training algorithm is proposed and its convergence property is justified.

First, we define the objective function as follows:

$$Q^{(t)}(\phi_x(y), \theta) = \frac{1}{|X_l^{(t)}|} \sum_{x\in X_l^{(t)}} G(\phi_x(y), \theta). \tag{30}$$

In the incremental mode, the cardinality $|X_l^{(t)}|$ of $X_l^{(t)}$ will be increased at each iteration. This objective function emphasizes the average objective value over the labeled data set $X_l^{(t)}$.

Let's start from $X^{(0)}$ and $\phi_x^{(0)}(y)$. We can get $\theta^{(0)}$ as the following:

$$\theta^{(0)} = \arg\max_{\theta} \sum_{x\in X_l^{(0)}} G(\phi_x^{(0)}(y), \theta). \tag{31}$$

The average value of objective function will be represented as $\lambda^{(0)}$:

$$\begin{aligned} \lambda^{(0)} &= Q^{(0)}(\theta^{(0)}, \phi_x^{(0)}(y)) \\ &= \frac{1}{|X_l^{(0)}|} \sum_{x\in X_l^{(0)}} G(\phi_x^{(0)}(y), \theta^{(0)}). \end{aligned} \tag{32}$$

Now let's assume we have $\phi_x^{(t-1)}(y)$, $\theta^{(t-1)}$, $X_l^{(t-1)}$ known. The objective function is represented as the following:

$$\begin{aligned}\lambda^{(t-1)} &= Q^{(t-1)}(\phi_x^{(t-1)}(y), \theta^{(t-1)}) \\ &= \frac{1}{|X_l^{(t-1)}|} \sum_{x \in X_l^{(t-1)}} G(\phi_x^{(t-1)}(y), \theta^{(t-1)}). \end{aligned} \tag{33}$$

According to the original source of data in both $X_l^{(t-1)}$ and $X_l^{(t-1)}$, the E-step (relabeling) will be categorized into three scenarios:

- $x \in X_l^{(0)}$

  For these observations, their label distributions will remain unchanged,

$$\phi_x^{(t)}(y) \quad = \quad \phi_x^{(0)}(y). \tag{34}$$

- $x \in X_l^{(t-1)} \setminus X_l^{(0)}$

  For these observations, the re-labeling strategy used in Case 3 will update their label distribution. Moreover, some of the data will be moved back to the unlabeled data set again if their objective value satisfies the following conditions. These removed data are represented as $\{x_{l \to u}^{(t-1)}\}$,

$$\begin{aligned} G(\phi_x^{(t)}(y), \theta^{(t-1)}) &\leq \alpha^{(t-1)} \\ &\leq \lambda^{(t-1)} \ (\forall x \in \{x_{l \to u}^{(t-1)}\}). \end{aligned} \tag{35}$$

  Overall, for all the observations in $X_l^{(t-1)} \setminus X_l^{(0)}$, the following inequality holds:

$$G(\phi_x^{(t)}(y), \theta^{(t-1)}) \ \geq \ G(\phi_x^{(t-1)}(y), \theta^{(t-1)}). \tag{36}$$

- $x \in X_u^{(t-1)}$

  For these observations, if their objective value satisfies the following conditions, they will be moved to the labeled data set,

$$\begin{aligned} G(\phi_x^{(t)}(y), \theta^{(t-1)}) &\geq \beta^{(t-1)} \\ &\geq \lambda^{(t-1)} \ (\forall x \ \in \{x_{u \to l}^{(t-1)}\}). \end{aligned} \tag{37}$$

After the relabeling process, the labeled data set is the following:

$$X_l^{(t)} = X_l^{(0)} \cup \{X_l^{(t-1)} \setminus X_l^{(0)} \setminus \{x_{l\to u}^{(t-1)}\}\} \cup \{x_{u\to l}^{(t-1)}\}. \tag{38}$$

The following inequalities hold:

$$\frac{1}{|X_l^{(t)}|} \sum_{x \in X_l^{(t)}} G(\phi_x^{(t)}(y), \theta^{(t-1)}) \quad \geq \quad \lambda^{(t-1)}. \tag{39}$$

Now the M-step (re-estimation) is to find the optimal $\theta^{(t)}$ over the labeled data set $X_l^{(t)}$ given that $\phi_x^{(t)}(y)$ is fixed:

$$\theta^{(t)} \quad = \quad \arg\max_{\theta} \sum_{x \in X_l^{(t)}} G(\phi_x^{(t)}(y), \theta). \tag{40}$$

It's clear that

$$\begin{aligned} \lambda^{(t)} &= \frac{1}{|X_l^{(t)}|} \sum_{x \in X_l^{(t)}} G(\phi_x^{(t)}(y), \theta^{(t)}) \\ &\geq \frac{1}{|X_l^{(t)}|} \sum_{x \in X_l^{(t)}} G(\phi_x^{(t)}(y), \theta^{(t-1)}) \\ &\geq \lambda^{(t-1)}. \end{aligned} \tag{41}$$

By this iterative procedure and appropriately chosen $\alpha^{(t)}$ and $\beta^{(t)}$, we can control the number of observations moved between X_u and X_l. Meanwhile, the objective function will increase until no more data will be moved between $X_l^{(\infty)}$ and $X_u^{(\infty)}$.

Some experiment results on audio event detection are shown in Figure 5 and Figure 6. For such an iterative procedure, the asymptotic convergence property is of concern. It's desirable to have some theoretical analysis for the asymptotic property of such a bootstrapping procedure. An alternative choice is to conduct asymptotic analysis through many experiments. These figures show the asymptotic convergence of the proposed method. The x-axis of both figures is the iteration number of the procedure. The y-axis of Figure 5 is the precision of the detected "applause/cheering"

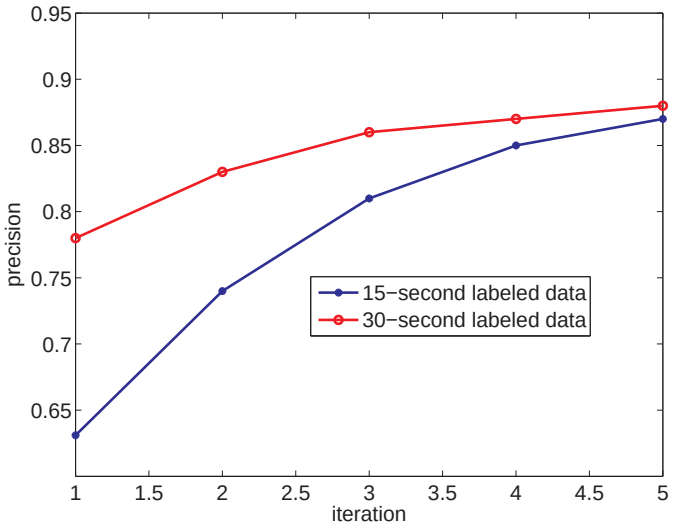

Figure 5: Semi-supervised audio event detection.

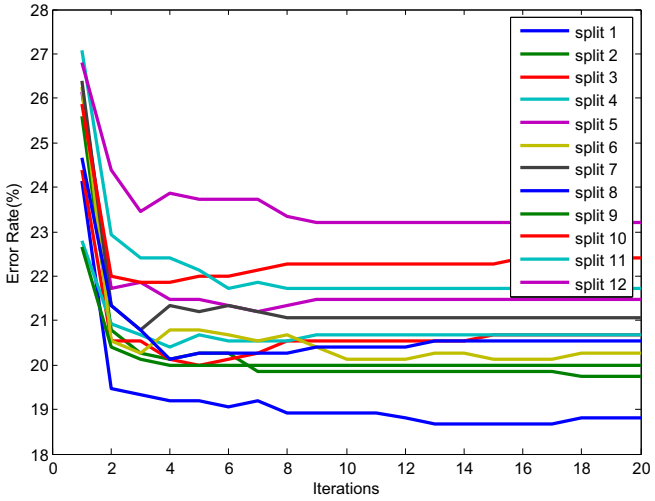

Figure 6: Semi-supervised audio event detection.

event. These figures indicate that with additional newly labeled positive and negative samples being added into the training set, the detection performance could be improved gradually.

CHAPTER III

INFORMATION INTEGRATION

Within the knowledge hierarchy of a detection-based framework, the higher level knowledge can be obtained by combining its lower level knowledge sources and level-specific information. For instance, the knowledge about phonemes could be obtained from the knowledge of distinctive features and phonological rules. Knowledge integration will be carried out at each level of a knowledge hierarchy, such as signal (sensor), feature, and decision level.

At the feature level, feature concatenation is a simple combination scheme, while at the decision level, some intuitive knowledge integration strategies are usually used, such as majority voting, weighted majority voting, and arithmetic average. Other fusion strategies such as the Dempster-Shafer theory has also been studied and employed for several decades [21].

In addition, detector interpolation (or randomized detectors) is another combination strategy. For instance, given a detector with a false positive probability 0.1 and true positive probability 0.2, and another detector with a false positive probability 0.25 and true positive probability 0.6, we could get detectors with the false positive probability in the range from 0.1 to 0.25 under the Neyman-Pearson criterion by detector randomization [26].

Knowledge integration can be conducted in either a parallel manner such as logistical combination of classifiers, or a sequential (cascade) fashion such as a decision tree. The basic principle is to gradually remove false alarms while keeping the true positive probability as high as possible by adding more evidence and constraints to the fusion procedure.

Knowledge integration seems similar to ensemble classifiers in some sense, such as bagging [7], boosting [93], stacking and so on. Generally, ensemble classifiers work on the same feature set through some kinds of re-sampling and weighting techniques for each classifier, while evidence fusion focuses on integrating heterogeneous knowledge sources from different detectors.

Knowledge integration could be formulated as follows: given a collection of detectors, $d_1, d_2, \cdots, d_n$, we are aiming to build a mapping $g(d_1, d_2, \cdots, d_n)$ from the input detectors to a higher level decision. Here, each d_i could either be a categorical classifier (output yes or no as in a decision rule) or a probabilistic classifier (output continuous score that can be calibrated into probability, as in ANNs) and g could be either an arithmetic function, a logical function, or a probabilistic model.

In the following sections, some commonly used combination schemes will be briefly discussed and then a novel discriminant function based combination approach is proposed.

3.1 Feature Concatenation and Model Combination

For signal and feature level integration, one simple approach is to concatenate the low-level features from different sources into a large vector, which is also called early fusion. However, from previous studies, it is known that some features are more informative than others and therefore scaling of each dimension is important. Otherwise, some dimensions will dominate the feature vectors. For instance, in a broadcast news video story segmentation system, the anchor shot is more significant than other pieces of video features. Another disadvantage of feature concatenation is that it is hard to incorporate heterogeneous information sources, e.g., combining numerical features with categorical features.

Model combination, or late fusion, is an alternative choice. By building a separate model for each knowledge source, the fusion is conducted at the decision level.

One disadvantage of model combination is the much higher computational complexity. For instance, we need to build separate decision trees, acoustic models, and transformation matrices for each feature stream in LVCSR systems.

These two combination schemes will be discussed in more detail in Chapter 4 for a comparative study on combination schemes for LVCSR systems.

3.2 Maximum Entropy Evidence Fusion

The maximum entropy (MaxEnt) model is a log-linear model that combines multiple binary features to approximate the posterior probability of an event (i.e., story boundary) given some evidence (e.g, audio, visual, or text information), which has also been studied for natural language processing and many other applications, [91],

$$P(\text{event}|\text{evidence}) = \frac{1}{Z_\lambda(\text{evidence})} \exp\left(\sum_{k=1}^{K} \lambda_k f_k(\text{event}, \text{evidence})\right). \quad (42)$$

Here, f_k are feature functions which are wrappers of the input evidence and the target event. The λ_k are the parameters of the MaxEnt model and $Z_\lambda(\text{evidence})$ is a normalization constant which ensures that probabilities sum to one. Since the normalization constant tends to contrast the ground truth label against any other label, this model is best suited for detection. In addition, it is ideally suited for combining detectors from different knowledge sources. A probabilistic detector can be quantized, thereby providing the ability to combine arbitrary detectors. The MaxEnt model is capable in dealing with heterogeneous features (acoustic, lexical, etc.) and sparse features.

The MaxEnt model has an equivalent formulation that maximizes the penalized log-likelihood as follows:

$$L = \sum_{i=1}^{l} P(y_i|x_i;\Lambda) - \sum_{k} \frac{\lambda_k^2}{2\sigma^2}, \quad (43)$$

where a regularization term has been added to control the generalization capability

of the model. The parameters $\Lambda = \{\lambda \cdots\}$ are trained to maximize penalized log-likelihood of a set of training labeled data $D_L = (x_1, y_1), \cdots, (x_l, y_l)$, where x is evidence and y is a event. The second sum is a Gaussian prior on the parameters to handle sparsity in the training data. For each $y \in Y$ there is a separate "default feature" that is always independent of x; this allows the model to represent a class prior probability. We use a single pooled variance σ^2 for all features. In practice, system performance seemed relatively insensitive over a large range of σ.

Each evidence from the data fits into a feature function that associates the attribute and a possible label as follows:

$$f_k(\text{event}, \text{evidence}) \quad = \quad \begin{cases} 1, & d_k(\text{evidence}) = \text{event} \\ 0, & \text{otherwise}, \end{cases}, \tag{44}$$

where a feature function takes a positive value if the attribute appears in the data, while a zero value if the attribute is not in the data. Each feature function carries a weight λ_k that gives the strength of that feature function for the proposed label. High positive weights indicate a good association between the feature and the proposed label. High negative weights indicate a negative association between the feature and the proposed label. Weights close to zero indicate the feature has little or no impact on the identity of the label.

The model parameters λ_k are trained using standard optimization techniques such as generalized iterative scaling (GIS), conjugate gradient (CG), or limited-memory BFGS (L-BFGS) methods [84].

The gradient is computed as follows:

$$\frac{\partial L}{\partial \lambda_k} \quad = \quad \sum_{i=1}^{k} f_k(x_i, y_i) - \sum_{y' \in Y} P(y_i|x_i) f(x_i, y') - \frac{\lambda_k}{\sigma^2}. \tag{45}$$

This model corresponds to an undirected graphical model (also known as a Markov random field (MRF)) in which the observed random variables x_i are connected by an

undirected edge to their corresponding random variables y_i, and the cliques of the graph are exactly these pairs, with clique potential $\Phi_i = \exp(\sum_k \lambda_k f_k(x_i, y_i))$.

A speaker identification experiment is conducted on the transcriptions of the broadcast news using the MaxEnt model. Our task aimed to assign a speaker name to each speaker cluster. And the input detectors are the N-gram pattern detectors, position detectors, gender detectors and acoustic speaker detectors. Experiments were carried out on two subsets of the broadcast news training datasets with testing set LDC97S44 and LDC97T22, and testing set LDC98S71 and LDC98T28. To increase the statistical significance of the results, 85-hours of data are used for testing.

Figure 7 shows the results of all the system configurations on the test set. With the same lexical trigger features, the MaxEnt system outperforms the N-gram system. We attribute the gain to the discriminative nature of the conditional training of parameters in the MaxEnt framework. In contrast, the N-gram system uses maximum likelihood estimates of parameters. The MaxEnt system jointly trains the feature combination parameters in contrast to the N-gram system where the rule combination is heuristic. With the position feature and gender information incorporated into the MaxEnt system, system performance is further improved. The results also show that the acoustic feature bears little incidence on overall performance. As mentioned before, acoustic detector can only help for the common speakers in the train and test sets. In our experimental setup, there are about 150 such common speakers. These common speakers account for 10% of the speakers in the test set and 30% of the test set in time-weighted proportion. An insufficient number of bins, or the relatively low discriminative power of the acoustic detector might be additional reasons for the lack of performance improvement.

We showed the benefit of the sound method for joint training of the parameters in the maximum entropy framework by demonstrating better performance than the state-of-the-art N-gram system while using the same set of lexical features. Our

experimental results show that at a fixed precision of 95%, our best MaxEnt system increases the recall from 38% to 67%.

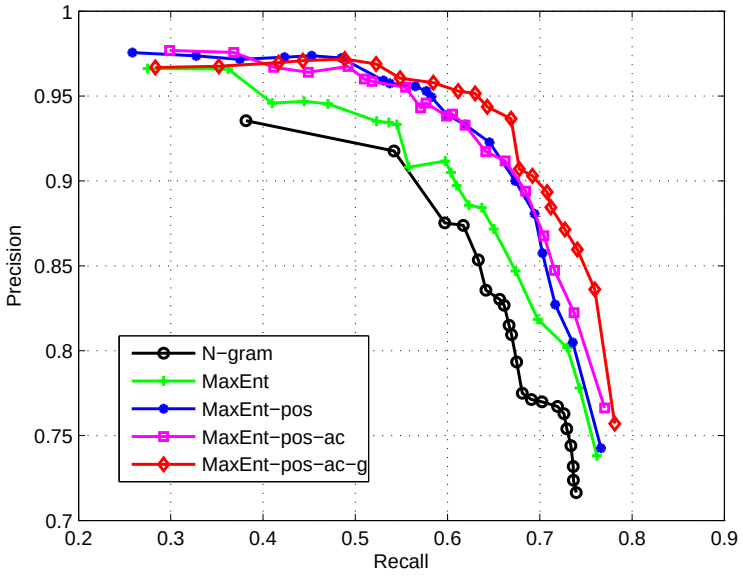

Figure 7: Maximum entropy fusion results.

3.3 Regularized MFoM for Integration

The MaxEnt model is a log-linear model that can be used to combine heterogeneous knowledge sources. In this section, a novel linear model is proposed to conduct information integration. The MaxEnt model is used to approximate the posterior probability, while the proposed discriminant function based approach aims at approximating any performance metric, such as error rate, recall, precision, and F1 measure.

In statistical decision theory, the optimal classifiers obtained by minimizing the Bayes risk are always defined in terms of a conditional probability $P(y|\mathbf{x})$ [23] [6] [38], where $\mathbf{x}$ is a feature vector and y is a class label. For most real-world classification problems, an exact knowledge of $P(y|\mathbf{x})$ is usually unavailable. There are two ways to handle the problem through using discriminative or generative models. Discriminative methods directly model the conditional probability distribution $P(y|\mathbf{x})$ with a

parametric form, such as logistic regression (LR) [66] and maximum entropy (MaxEnt) model [4]. In contrast, generative methods assume the data is generated by a joint probabilistic distribution $P(\mathbf{x}, y)$, which is usually factorized into a product of two terms: a prior probability $P(y)$ such as a binomial or multinomial distribution, and a class conditional probability distribution, $P(\mathbf{x}|y)$. These generative models include naive Bayes models, Gaussian mixture models (GMM) and hidden Markov models (HMM) [87]. The posterior distribution $P(y|\mathbf{x})$ is derived from $P(\mathbf{x}, y)$ by the Bayes theorem. The parameters of discriminative and generative models are typically estimated by either the maximum likelihood (ML) or maximum *a posterior* (MAP) estimation criteria [60] [10].

Two aspects of these statistical learning and inference schemes need to be elaborated as follows: First, the success of these statistical modeling schemes depends on the correctness of the model assumptions. In case the actual training and testing data do not support the model assumptions needed in generative methods as we lack complete knowledge of the parametric form of the underlying distributions, system performance can no longer be optimal. Discriminant function based approaches are more flexible, because it directly maps feature vector, $\mathbf{x}$, to its class label, y. Discriminant functions are used to partition the sample space into several non-overlapping regions. We only specify a parametric form of the discriminant functions $g_i(\mathbf{x})$ for each class i. Support vector machine (SVM) is a linear discriminant function (LDF) based classifier [8]. Discriminant functions can also be regarded as a generalization of the conditional distribution if we regard $P(y|\mathbf{x})$ as a discriminant function and generative models can be transformed to discriminant functions by the Bayes rule.

Second, the performance metrics used in system evaluation on unseen test data are often different from the ML and MAP criteria used in model training. For instance, word error rate is usually adopted in the evaluation of automatic speech recognition

systems [88]. Whereas, recall, precision, and F1 measure are often employed in information retrieval (IR) [76] and video story segmentation systems [68]. On the other hand, automatic image annotation systems are usually evaluated by mean average precision [75], and speaker verification systems are commonly measured by equal error rate and detection cost functions [77]. So in the last two decades, there are many efforts devoted to making model training criteria consistent with the performance evaluation metrics. Minimum classification error (MCE) learning was proposed to directly minimize the objective function approximating the empirical errors on the training set by embedding discriminant functions and decision rules into parameter learning criteria [48]. The MCE criterion has been successfully used in speech recognition [47] [94], speaker recognition [69], and many other applications, where the generative models, such as hidden Markov models, are discriminatively trained by optimizing the MCE criterion. Maximum figure-of-merit (MFoM) learning [31] generalizes the MCE criterion to cover more performance metrics such as recall, precision, F1 measure, and the area under the receiver operating characteristic (ROC) curve [30] [26]. MFoM learning have been successfully used in many applications, such as text categorization [31], automatic image annotation and video segmentation [68].

In this section, we propose a regularized maximum figure-of-merit (rMFoM) approach to supervised learning within the framework of Tikhonov regularization [100]. The rMFoM learning criterion has all the advantages of the original MFoM learning, i.e., it can be tailored to task-specific performance metrics for practical applications. Furthermore, it can deal with the problem of insufficient training data because it is less sensitive to potential data variations [33] [31].

The rMFoM approach is also successfully extended to the general semi-supervised learning (SSL) scenarios [13]. Recently, SSL has been investigated extensively in text categorization (TC) [83], web mining and computational linguistics [1], where

labeled data are often difficult and expensive to obtain. The semi-supervised rMFoM approach provides a theoretical justification of the commonly used self-training algorithm [83] [108]. Both quasi-Newton and trust region Newton methods [84] are applied to this non-convex optimization problem and several implementation issues are discussed in detail.

We evaluate supervised rMFoM learning on the Reuters-21578 text categorization task. For semi-supervised rMFoM learning, we also conducted experiments on other two text categorization datasets. We focus our study on binary classification with linear discriminant functions. The results clearly show that the performance of each classifier is consistent between the training and evaluation stage, and the classifier is able to obtain the best chosen metric on the testing data.

3.3.1 Maximum Figure-of-merit Learning

Let's start with a brief review of the maximum figure-of-merit (MFoM) approach to supervised learning [33]. Suppose we are given a dataset, $X = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{M}$, consisting of M labeled training samples, here $\mathbf{x}^{(i)} \in \mathbb{R}^n$ is a n-dimension feature vector and $y^{(i)} \in Y = \{1, 2, \cdots, K\}$ is its class label. Discriminant function approaches do not require knowledge of the forms of underlying probability density functions of the joint distribution, $p(\mathbf{x}, y)$, in generative models or the conditional distribution, $p(y|\mathbf{x})$, in discriminative models. We only assume we have a parametric form of the discriminant functions, $g_j(\mathbf{x}; \mathbf{w}_j)$, parameterized by $\mathbf{w}_j$ for each class $j \in Y$. An intuitive decision rule for discriminant function based classification is to maximize over all the class scores of an instance $\mathbf{x}$ as follows:

$$\hat{C}(\mathbf{x}) \quad = \quad \arg\max_{j \in Y} g_j(\mathbf{x}; \mathbf{w}_j), \tag{46}$$

where $\hat{C}(\mathbf{x})$ is the predicted class label for $\mathbf{x}$.

For any labeled instance $\mathbf{x} \in C_j$, a correct classification means $\hat{C}(\mathbf{x}) = C_j$. To quantify the classification performance of an instance $\mathbf{x} \in C_j$, a class misclassification

measure $d_j(\mathbf{x})$ is defined for each class C_j by:

$$d_j(\mathbf{x}) = -g_j(\mathbf{x};\mathbf{w}_j) + \left\{ \frac{1}{K-1} \Big[\sum_{i=1, i\neq j}^{K} g_i^{\eta}(\mathbf{x};\mathbf{w}_i) \Big] \right\}^{1/\eta}, \tag{47}$$

where the second term is the geometric mean of the scores from all competing classes and η is a positive constant. When η approaches $+\infty$, the second term becomes the maximal score from all competing classes,

$$d_j(\mathbf{x}) \;\; = \;\; -g_j(\mathbf{x};\mathbf{w}_j) + \max_{i\neq j} g_i(\mathbf{x};\mathbf{w}_i). \tag{48}$$

If $d_j(\mathbf{x})$ is positive, it means that $\mathbf{x} \in C_j$ is misclassified as one of the other classes. In this case, $d_j(\mathbf{x})$ can also be regarded as the negative of a generalized functional margin. It quantifies the separation between class C_j and all other competing classes. The misclassification measure for each $\mathbf{x} \in C_j$ can be embedded into a sigmoid function to approximate the error count of the classification process, e.g., 0-1 for correct or incorrect decisions as follows:

$$\ell_j(\mathbf{x}) \;\; = \;\; \frac{1}{1+\exp\big[-\alpha\big(d_j(\mathbf{x})+\beta_j\big)\big]}, \tag{49}$$

where α controls the steepness of the approximation curve and β_j control the position of the decision boundary for class C_j.

The confusion matrix (a.k.a. contingency table) for class C_j consisting of true positive (TP), false positive (FP), false negative (FN) and true negative (TN) is shown in Table 5 in terms of a smooth continuous and differentiable loss function, $\ell_j(\mathbf{x})$.

Table 5: Confusion matrix

	$\hat{C}(\mathbf{x}) = C_j$	$\hat{C}(\mathbf{x}) \neq C_j$
$\mathbf{x} \in C_j$	$\mathrm{TP}_j \approx \sum_{\mathbf{x}\in C_j} \big(1-\ell_j(\mathbf{x})\big)$	$\mathrm{FN}_j \approx \sum_{\mathbf{x}\in C_j} \ell_j(\mathbf{x})$
$\mathbf{x} \notin C_j$	$\mathrm{FP}_j \approx \sum_{\mathbf{x}\notin C_j} \ell_j(\mathbf{x})$	$\mathrm{TN}_j \approx \sum_{\mathbf{x}\notin C_j} \big(1-\ell_j(\mathbf{x})\big)$

For a specific class C_j, several performance metrics, like error rate, precision, recall and F1 can be calculated as shown in Table 6. Here, TP, FP, FN and TN are approximated with the loss function for each training sample.

Table 6: Performance metrics

Error	$\mathrm{E}_j = \dfrac{\mathrm{FP}_j + \mathrm{FN}_j}{\mathrm{TP}_j + \mathrm{TN}_j + \mathrm{FP}_j + \mathrm{FN}_j}$
Recall	$\mathrm{R}_j = \dfrac{\mathrm{TP}_j}{\mathrm{TP}_j + \mathrm{FN}_j}$
Precision	$\mathrm{P}_j = \dfrac{\mathrm{TP}_j}{\mathrm{TP}_j + \mathrm{FP}_j}$
F1	$\mathrm{F1}_j = \dfrac{2\mathrm{TP}_j}{\mathrm{FP}_j + \mathrm{FN}_j + 2\mathrm{TP}_j}$

By this way, these performance measures are differentiable functions with respect to the model parameters, and they can therefore be directly optimized. The basic idea of MFoM learning approach is to directly optimize the abovementioned performance metrics with respect to discriminant function parameters. It allows quite a bit of flexibility in choosing the discriminant functions for each class. For instance, the discriminant functions $g_j(\mathbf{x}; \mathbf{w}_j)$ could be linear discriminant functions (LDF) if the data is fairly linearly separable in the feature space. LDFs are essentially minimum error Bayesian classifiers, which assume that the class conditional probability densities are multivariate Gaussians with equal covariance matrices for every class. Otherwise, some nonlinear discriminant functions like quadratic discriminant functions, or more complicated probabilistic discriminant functions like log-likelihood function of Gaussian mixture models (GMM) and hidden Markov models (HMM) could be employed. An advantage of MFoM learning is that it can be applied to both binary and multi-class classification problems.

3.3.2 Supervised rMFoM Learning

The MCE criterion with LDFs can be regarded as a single layer artificial neural network (ANN) with a sigmoid transfer function. Discriminative training methods

with the MCE criterion are very effective in practice. However, they are prone to overfitting [85] [89] and smoothing techniques such as I-smoothing are used to alleviate overfitting [85]. We believe similar phenomena can be observed in MFoM learning as well. For instance, suppose we are using MFoM learning with LDFs, $g(\mathbf{x};\mathbf{w},\beta) = \mathbf{w}^T\mathbf{x}+\beta$, for binary classification problems, if we scale up the parameters $\mathbf{w}$ to $2\mathbf{w}$, the classification hyperplane will not change. However, the value of the objective function in Eq. (49) will decrease. So it is necessary to add some penalty or regularization terms to the original MFoM formulation in order to achieve better generalization and stability.

An intuitive way is to add a norm constraint on the parameters of the classification hyperplane as follows:

$$\begin{aligned} \min_{\mathbf{w},\beta} f(\mathbf{w},\beta) &= \begin{cases} \frac{1}{M}(\mathrm{FP}+\mathrm{FN}), & \mathrm{MCE} \\ -\frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}, & \mathrm{F1} \end{cases} \\ \text{subject to} \quad & \|\mathbf{w}\|_2 = 1. \end{aligned} \tag{50}$$

Here the nonlinear equality constraint $\|\mathbf{w}\|_2 = 1$ restricts the feasible set of $\mathbf{w}$ to be the non-convex surface of the unit sphere. In the following sections of this paper, we will derive detailed algorithms for the two typical MFoM criteria: minimum classification error (MCE) and maximum F1 as shown in Eq. (50). Using the method of Lagrange multipliers, the Lagrangian can be formed as follows:

$$\min_{\mathbf{w},\beta,\lambda} L(\mathbf{w},\beta,\lambda) = f(\mathbf{w},\beta) + \lambda(\mathbf{w}^T\mathbf{w}-1), \tag{51}$$

where λ is a Lagrange multiplier.

From another perspective, this problem can be reformulated within the framework of Tikhonov regularization [100]. In statistics [38] and machine learning, regularization is usually used to prevent overfitting, e.g., ridge regression [6], Lasso [99], and ℓ_2-norm regularization in support vector machines [95]. Tikhonov proposed a class of

Tikhonov regularizers that are formulated as follows:

$$\min_{h\in\mathcal{H}}\Big[\ell\big(h(\mathbf{X}),\mathbf{y}\big)+\frac{\lambda}{2}\|h\|_{\mathcal{H}}^2\Big],\quad \lambda>0, \tag{52}$$

where h is a prediction function in the functional space $\mathcal{H}$, $\ell\big(h(\mathbf{X}),\mathbf{y}\big)$ is a loss function, and $\|h\|_{\mathcal{H}}$ is the norm in the functional space $\mathcal{H}$. Tikhonov regularization seeks a function which simultaneously exhibits a small empirical loss on the training set and a small norm in a reproducing kernel Hilbert space. Of course, other types of regularization like ℓ_1-norm regularization can be used in other scenarios, which does both variable selection and norm shrinking and can produce sparse parameter vectors as well [55].

To simplify the derivation of related algorithms, in this paper, our discussion of the regularized MFoM learning algorithm will focus on binary classification using linear discriminant functions, i.e., $Y=\{1,-1\}$. Each feature vector and the LDF parameters are augmented with an additional dimension like this: $\mathbf{x}^{(i)}\leftarrow[\mathbf{x}^{(i)},1]$, and $\mathbf{w}^T\leftarrow[\mathbf{w}^T,\beta]$. It can be extended to multi-class classification problems as well [31]. For binary classification, the whole training set is explicitly partitioned into two parts, $X=\{X_{\text{pos}},X_{\text{neg}}\}$ and the loss functions is represented as follows:

$$\ell(\mathbf{x}^{(i)};\mathbf{w}) \;=\; \frac{1}{1+\exp\left(\alpha y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)}\right)}. \tag{53}$$

In this paper, as a simple case of the Tikhonov regularizer for linear discriminant functions, we employed the following regularization terms:

$$f(\mathbf{w})=\begin{cases}\frac{\lambda}{2}\mathbf{w}^T\mathbf{w}+\frac{1}{M}(\text{FP}+\text{FN}), & \text{MCE}\\ \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}-\frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}}, & \text{F1}\end{cases} \tag{54}$$

In the case of MCE learning, the gradient and Hessian matrix of the regularized MFoM objective function $f(\mathbf{w})$ can be represented as follows:

$$\nabla_{\mathbf{w}}f(\mathbf{w}) \;=\; \lambda\mathbf{w}+\mathbf{X}^T\mathbf{c}, \tag{55}$$

$$\nabla^2_{\mathbf{w}}f(\mathbf{w}) \;=\; \lambda\mathbf{I}+\mathbf{X}^T\mathbf{D}\mathbf{X}. \tag{56}$$

Here $\mathbf{X}^T = [\mathbf{x}^{(1)}\ \mathbf{x}^{(2)}\ \cdots\ \mathbf{x}^{(M)}]$ is the data matrix and $\mathbf{c}$ is a vector with each entry

$$\mathrm{c}_i = \frac{\alpha y^{(i)}}{M}\ell(\mathbf{x}^{(i)})\big(\ell(\mathbf{x}^{(i)}) - 1\big).$$

$\mathbf{I}$ is the identity matrix and $\mathbf{D}$ is a diagonal matrix with

$$\mathbf{D}_{ii} = \frac{\alpha^2}{M}\ell(\mathbf{x}^{(i)})\big(\ell(\mathbf{x}^{(i)}) - 1\big)\big(2\ell(\mathbf{x}^{(i)}) - 1\big).$$

In the case of maximizing the F1 measure, the gradient and Hessian matrix of the regularized MFoM objective function are the following:

$$\nabla_{\mathbf{w}} f(\mathbf{w}) = \lambda\mathbf{w} + \mathbf{X}^T\mathbf{v}, \tag{57}$$

$$\nabla^2_{\mathbf{w}} f(\mathbf{w}) = \lambda\mathbf{I} + \mathbf{X}^T(\mathbf{K}^T\mathbf{S}\mathbf{K} + \mathbf{Q})\mathbf{X}. \tag{58}$$

Here $\mathbf{v}$ is a vector with

$$\mathbf{v}_i = \mathrm{r}_i\alpha\ell(\mathbf{x}^{(i)})\big(\ell(\mathbf{x}^{(i)}) - 1\big),$$

and r_i is defined as follows:

$$\mathrm{r}_i = \begin{cases} \frac{2(\mathrm{TP}+\mathrm{FP}+\mathrm{FN})}{(\mathrm{FP}+\mathrm{FN}+2\mathrm{TP})^2}, & \mathbf{x}^{(i)} \in \mathrm{X}_{\mathtt{pos}} \\ \frac{-2\mathrm{TP}}{(\mathrm{FP}+\mathrm{FN}+2\mathrm{TP})^2}, & \mathbf{x}^{(i)} \in \mathrm{X}_{\mathtt{neg}} \end{cases}$$

Both $\mathbf{K}$ and $\mathbf{Q}$ are diagonal matrices, and $\mathbf{S}$ is a symmetric matrix whose elements are defined in Table 7.

$$\mathbf{K}_{ii} = \alpha\ell(\mathbf{x}^{(i)})\big(\ell(\mathbf{x}^{(i)}) - 1\big)$$

$$\mathbf{Q}_{ii} = \mathrm{r}_i\alpha^2 y^{(i)}\ell(\mathbf{x}^{(i)})\big(\ell(\mathbf{x}^{(i)}) - 1\big)\big(2\ell(\mathbf{x}^{(i)}) - 1\big)$$

Table 7: Matrix $\mathbf{S}$

$\mathbf{S}_{ij}$	$\mathbf{x}^{(j)} \in \mathrm{X}_{\mathtt{pos}}$	$\mathbf{x}^{(j)} \in \mathrm{X}_{\mathtt{neg}}$
$\mathbf{x}^{(i)} \in \mathrm{X}_{\mathtt{pos}}$	$\frac{4(\mathrm{TP}+\mathrm{FP}+\mathrm{FN})}{(\mathrm{FP}+\mathrm{FN}+2\mathrm{TP})^3}$	$\frac{2(\mathrm{FP}+\mathrm{FN})}{(\mathrm{FP}+\mathrm{FN}+2\mathrm{TP})^3}$
$\mathbf{x}^{(i)} \in \mathrm{X}_{\mathtt{neg}}$	$\frac{2(\mathrm{FP}+\mathrm{FN})}{(\mathrm{FP}+\mathrm{FN}+2\mathrm{TP})^3}$	$\frac{-4\mathrm{TP}}{(\mathrm{FP}+\mathrm{FN}+2\mathrm{TP})^3}$

Table 8: Three learning frameworks for binary classification

Algorithm	$\min_{\mathbf{w}} f(\mathbf{w})$
SVM	$\frac{\lambda}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{M}\sum_{i=1}^{M}(1 - y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)})_+$
rLR	$\frac{\lambda}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{M}\sum_{i=1}^{M}\log\big(1 + \exp(-y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)})\big)$
rMFoM	$\frac{\lambda}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{M}\sum_{i=1}^{M}\frac{1}{1 + \exp(\alpha y^{(i)}\mathbf{w}^T\mathbf{x}^{(i)})}$

Linear support vector machine (SVM) [8], regularized logistic regression (rLR) [66], and regularized maximum figure-of-merit (rMFoM) learning with linear discriminant functions are all binary classifiers. Their objective functions are summarized in Table 8. Although derived from different motivations, they share some similarities in the form of their objective functions: summation of an empirical loss function and a regularization term. Nevertheless, their differences lie in the choice of the loss functions as shown in Figure 8: a hinge loss function for SVM, a logistic loss function for rLR, and a sigmoid loss function for rMFoM. An advantage of linear SVM and rLR is that their objective functions are convex function in term of model parameters and will converge to a global minimizer. However, the Hessian matrices of the rMFoM objective functions are usually indefinite and so their objective functions are non-convex functions, which means that it can only converge to a local minimum. However, only the rMFoM criteria are consistent with defining the model training criteria and evaluation metrics.

For the regularization term in the regularized logistic regression in Table 8, it can be interpreted as posing a Gaussian prior on the parameters $\mathbf{w}$. As for the linear SVM, the regularization term is related to its functional margin around the hyperplane separating the two classes. On the other hand, for rMFoM learning, we don't have a probabilistic interpretation on the regularization term. Generally speaking, when a regularization term is added to the original objective function, it works as a constraint

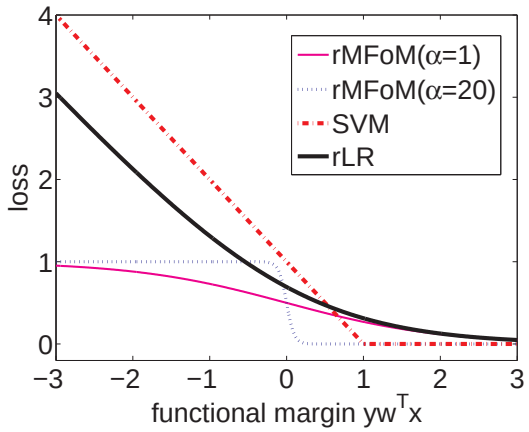

Figure 8: Comparison of loss functions.

in the optimization problem and it conveys some kind of prior knowledge about the target function to be learnt.

3.3.3 Semi-supervised rMFoM Learning

Table 9: Four terms in a contingency table

$$\text{TP} \approx \sum_{\mathbf{x}^{(i)} \in \mathrm{X_{pos}}} \left(1 - \ell_+(\mathbf{x}^{(i)}; \mathbf{w})\right) + \gamma \sum_{\mathbf{x}^{(j)} \in X_u} z^{(j)} \left(1 - \ell_+(\mathbf{x}^{(j)}; \mathbf{w})\right)$$

$$\text{FN} \approx \sum_{\mathbf{x}^{(i)} \in \mathrm{X_{pos}}} \ell_+(\mathbf{x}^{(i)}; \mathbf{w}) + \gamma \sum_{\mathbf{x}^{(j)} \in X_u} z^{(j)} \ell_+(\mathbf{x}^{(j)}; \mathbf{w})$$

$$\text{FP} \approx \sum_{\mathbf{x}^{(i)} \in \mathrm{X_{neg}}} \ell_-(\mathbf{x}^{(i)}; \mathbf{w}) + \gamma \sum_{\mathbf{x}^{(j)} \in X_u} (1 - z^{(j)}) \ell_-(\mathbf{x}^{(j)}; \mathbf{w})$$

$$\text{TN} \approx \sum_{\mathbf{x}^{(i)} \in \mathrm{X_{neg}}} \left(1 - \ell_-(\mathbf{x}^{(i)}; \mathbf{w})\right) + \gamma \sum_{\mathbf{x}^{(j)} \in X_u} (1 - z^{(j)}) \left(1 - \ell_-(\mathbf{x}^{(j)}; \mathbf{w})\right)$$

Now we extend the discussion of rMFoM learning to semi-supervised learning (SSL). In an SSL scenario, the whole training set X is divided into two subsets, X_l and X_u, where $X_l = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{M}$ is the original labeled data and $X_u = \{\mathbf{x}^{(j)}\}_{j=1}^{U}$, $U \gg M$ is the unlabeled data and $y^{(i)} \in \{-1, 1\}$ is the class label. To fully leverage the abundance of unlabeled data, almost all SSL algorithms make some assumptions about the underlying data distribution, such as smoothness and manifold assumptions [13] [108]. Self-training [83] is the most intuitive SSL algorithm. It is an iterative wrapper learning setup. At first, part or all of the unlabeled data are labeled by the

current model. Then, the classifier parameters are re-estimated with all the newly labeled data assuming the labels are correct. This procedure repeats itself until convergence or some stop criteria are met. Other SSL algorithms include generative model based SSL methods, semi-supervised or transductive support vector machine (TSVM) based on a cluster assumption that there is a wide margin in a kernel induced feature space between unlabeled data from different classes [46], graph-based SSL algorithms assume that labeled and unlabeled examples are connected by a graph, where edges represent similarity between examples [108].

For an unlabeled sample, $\mathbf{x} \in X_u$, we need to explicitly represent its loss with respect to the both positive and negative classes. So we use $\ell_+(\mathbf{x}^{(i)}; \mathbf{w})$ as the loss function if $\mathbf{x}^{(i)}$ is a positive instance and $\ell_-(\mathbf{x}^{(i)}; \mathbf{w})$ if $\mathbf{x}^{(i)}$ is a negative instance:

$$\ell_+(\mathbf{x}^{(i)}; \mathbf{w}) = \frac{1}{1 + \exp\left(\alpha \mathbf{w}^T \mathbf{x}^{(i)}\right)}, \text{ and} \tag{59}$$

$$\ell_-(\mathbf{x}^{(i)}; \mathbf{w}) = \frac{1}{1 + \exp\left(-\alpha \mathbf{w}^T \mathbf{x}^{(i)}\right)} = 1 - \ell_+^{(i)}. \tag{60}$$

For each unlabeled instance $\mathbf{x}^{(j)} \in X_u$, we assume there is a missing class label $z^{(j)} \in \{0, 1\}$ associated with it. Here we use $\mathbf{z} = [z^{(1)}, z^{(2)}, \cdots, z^{(U)}]^T$ as the missing labels for all unlabeled instances X_u. With the help of the missing label vector $\mathbf{z}$, it is straightforward to extend the four terms of a confusion matrix in Table 5 to incorporate unlabeled data in terms of the two explicit loss functions $\ell_+(\mathbf{x}^{(i)}; \mathbf{w})$ and $\ell_-(\mathbf{x}^{(i)}; \mathbf{w})$. The confusion matrix in the semi-supervised learning scenario is shown in Table 9, where γ is a weight assigned to unlabeled data.

In the case of minimum classification error (MCE) learning, the proposed objective

function can be represented as follows:

$$\min_{\mathbf{w},\mathbf{z}} f(\mathbf{w},\mathbf{z}) = \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{M}\Big[\sum_{\mathbf{x}_i\in \mathrm{X_{pos}}} \ell_+^{(i)} + \sum_{\mathbf{x}_i\in \mathrm{X_{neg}}} \ell_-^{(i)}\Big]$$
$$+ \quad \frac{\gamma}{U}\sum_{\mathbf{x}_j\in X_u}\big(z^{(j)}\ell_+^{(i)} + (1-z^{(j)})(1-\ell_+^{(i)})\big)$$
$$\text{subject to} \quad \frac{1}{U}\sum_{j=1}^{U} z^{(j)} = r \text{ and } z^{(j)} \in \{0,1\}. \tag{61}$$

Here we impose a constraint on the missing label vector $\mathbf{z}$ by restricting the ratio of positive instances in X_u to be a prefixed constant r. Similar constraints have been proposed in several other SSL algorithms such as transductive support vector machine (TSVM) [13] [95].

The proposed objective function involves a mixed optimization problem with both continuous variable $\mathbf{w}$ and discrete variable $\mathbf{z}$. This objective function can be optimized with a coordinate descent algorithm [84]. First, given a fixed $\mathbf{z}_t$, the optimal $\mathbf{w}_t$ can be obtained by optimizing the unconstrained objective function. Secondly, given a fixed $\mathbf{w}_t$, a better label assignment $\mathbf{z}_{t+1}$ could be obtained by optimizing the constrained objective function with respect to $\mathbf{z}$.

The gradient and Hessian matrix of the objective function $f(\mathbf{w};\mathbf{z}_t)$ can be evaluated as follows:

$$\nabla_{\mathbf{w}} f(\mathbf{w};\mathbf{z}_t) = \lambda\mathbf{w} + \mathbf{X}^T\mathbf{c}, \tag{62}$$
$$\nabla^2_{\mathbf{w}} f(\mathbf{w};\mathbf{z}_t) = \lambda\mathbf{I} + \mathbf{X}^T\mathbf{D}\mathbf{X}. \tag{63}$$

Here $\mathbf{X}^T = [\mathbf{x}^{(1)}\ \mathbf{x}^{(2)}\ \cdots\ \mathbf{x}^{(M)}\ \mathbf{x}^{(M+1)}\ \cdots\ \mathbf{x}^{(M+U)}]$ is the data matrix and $\mathbf{c}$ is a vector with

$$\mathrm{c}_i = \begin{cases} \frac{\alpha y^{(i)}}{M}\ell_+^{(i)}(\ell_+^{(i)}-1), & \mathbf{x}^{(i)} \in \mathrm{X}_l \\ \frac{\alpha\gamma(2z_t^{(i)}-1)}{U}\ell_+^{(i)}(\ell_+^{(i)}-1), & \mathbf{x}^{(i)} \in \mathrm{X}_u \end{cases}$$

$\mathbf{I}$ is the identity matrix, $\mathbf{D}$ is a diagonal matrix with

$$\mathbf{D}_{ii} = \begin{cases} \frac{\alpha^2}{M}\ell_+^{(i)}(\ell_+^{(i)}-1)(2\ell_+^{(i)}-1), & \mathbf{x}^{(i)} \in \mathrm{X}_l \\ \frac{\alpha^2\gamma(2z_t^{(i)}-1)}{U}\ell_+^{(i)}(\ell_+^{(i)}-1)(2\ell_+^{(i)}-1), & \mathbf{x}^{(i)} \in \mathrm{X}_u \end{cases}$$

With a fixed $\mathbf{w}_t$, the optimal label assignment vector $\mathbf{z}_{t+1}$can be obtained by the following 0-1 integer programming problem.

$$\begin{aligned} \min_{\mathbf{z}} f(\mathbf{z}; \mathbf{w}_t) &= \sum_{\mathbf{x}_j \in X_u} z^{(j)}(2\ell_+^{(i)} - 1) \\ \text{subject to} \quad & \frac{1}{U}\sum_{j=1}^{U} z^{(j)} = r \text{ and } z^{(j)} \in \{0, 1\}. \end{aligned} \tag{64}$$

Although integer programming problems are often difficult, the solution to this problem is quite straightforward. First, all the loss value of unlabeled data $\{\ell_+^{(j)}\}_{j=1}^{U}$ are sorted in a descent order. Then assign $z_{t+1}^{(j)} = 1$ for these instances whose loss value $\ell_+^{(j)}$ is one of the top ur smallest in the sorted list. For all the other instances, their $z_{t+1}^{(j)}$ will be 0.

This procedure provides us with a theoretical justification of the widely-used self-training algorithm in semi-supervised learning [13]. It coincides with our intuition for self-training. Actually, this procedure can be regarded as an extension of the EM algorithm [22] to discriminant function based semi-supervised learning.

An alternative strategy to the abovementioned integer programming problem is to use deterministic annealing [90] [95], which has been successfully applied to many combinatorial optimization problems, such as graph partitioning and clustering [90]. Deterministic annealing is a commonly used relaxation technique that can be used to relax discrete variables in combinatorial optimization problems to continuous variables and the original objective function is augmented by an entropy term $H(\mathbf{p})$as

follows:

$$\begin{aligned}
&\min_{\mathbf{w},\mathbf{p}} f(\mathbf{w},\mathbf{p}) = \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{M}\Big[\sum_{\mathbf{x}_i\in \mathrm{X_{pos}}} \ell_+^{(i)} + \sum_{\mathbf{x}_i\in \mathrm{X_{neg}}} \ell_-^{(i)}\Big] \\
+ \quad & \frac{\gamma}{U}\sum_{\mathbf{x}_j\in X_u}\Big(p^{(j)}\ell_+^{(i)} + (1-p^{(j)})(1-\ell_+^{(i)})\Big) \\
+ \quad & \frac{T}{U}\sum_{\mathbf{x}_j\in X_u}\Big(p^{(j)}\log\left(p^{(j)}\right) + (1-p^{(j)})\log\left(1-p^{(j)}\right)\Big) \\
& \text{subject to} \quad \frac{1}{U}\sum_{j=1}^{U} p^{(j)} = r \text{ and } p^{(j)}\in[0,1].
\end{aligned} \tag{65}$$

Here we replace the discrete variable $\mathbf{z}$ with a continuous variable $\mathbf{p}$, where $\mathbf{p} = [p^{(1)}, p^{(2)}, \cdots, p^{(U)}]^T$ and can be regarded as the probability of being positive for each instance of X_u and $p^{(j)} \in [0,1]$. Here T is the temperature that evolves according to a pre-specified annealing scheme. When T approach 0, the augmented objective function will become the original objective function.

When the label assignment vector $\mathbf{p}_t$ is given, the optimization with respect to $\mathbf{w}$ is the same as the procedure described in Section 3.3.3 by replacing $z_t^{(j)}$ by $p_t^{(j)}$.

With a fixed $\mathbf{w}_t$, the optimal label assignment vector $\mathbf{p}_{t+1}$can be obtained by optimizing the following constrained problem:

$$\begin{aligned}
&\min_{\mathbf{p}} f(\mathbf{p};\mathbf{w}_t) = \frac{\gamma}{U}\sum_{\mathbf{x}_j\in X_u} p^{(j)}(2\ell_+^{(i)} - 1) \\
+ \quad & \frac{T}{U}\sum_{\mathbf{x}_j\in X_u}\Big(p^{(j)}\log\left(p^{(j)}\right) + (1-p^{(j)})\log\left(1-p^{(j)}\right)\Big) \\
& \text{subject to} \quad \frac{1}{U}\sum_{j=1}^{U} p^{(j)} = r \text{ and } p^{(j)}\in[0,1].
\end{aligned} \tag{66}$$

By introducing a Lagrange multiplier ν for the equality constraint in Eq. 66, $\mathbf{p}_{t+1}$ can be solved in a closed-form. In order to compute $\mathbf{p}_{t+1}$ we need to know the value of the multiplier ν by solving a scalar nonlinear equation as shown in Eq. (68) in the following, which can be solved using a combination of interval bisection, linear interpolation, and inverse quadratic interpolation [84]. Figure 9 shows an example

of the histogram of $\ell_+^{(i)}$ for some unlabeled data and the corresponding nonlinear equation at two different annealing temperature T.

$$p_{t+1}^{(j)} = \frac{1}{1+\exp\left(\frac{\gamma(2\ell_+^{(j)}-1)-\nu}{T}\right)}, \tag{67}$$

$$\frac{1}{U}\sum_{\mathbf{x}_j \in X_u} \frac{1}{1+\exp\left(\frac{\gamma(2\ell_+^{(j)}-1)-\nu}{T}\right)} = r. \tag{68}$$

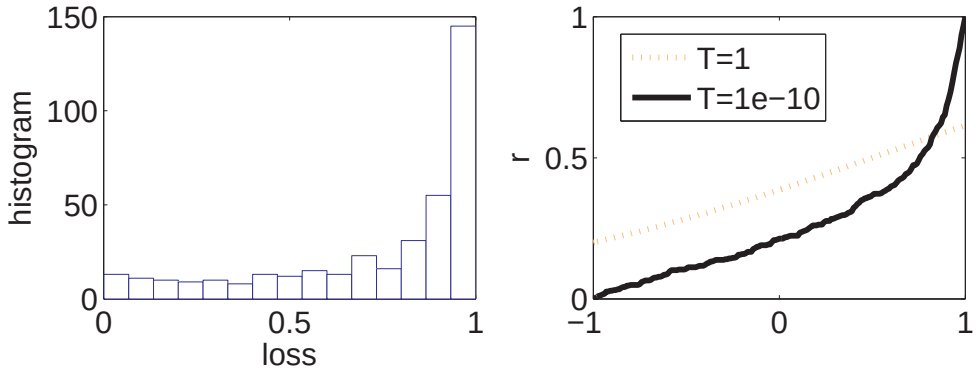

Figure 9: Nonlinear scalar equation for ν.

In the case of maximizing the F1 measure, the proposed objective function can be represented as follows:

$$\begin{aligned} \min_{\mathbf{w},\mathbf{p}} f(\mathbf{w},\mathbf{p}) &= \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} - \frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}} \\ &+ \frac{T}{U}\sum_{\mathbf{x}_j \in X_u}\left(p^{(j)}\log\left(p^{(j)}\right) + \left(1-p^{(j)}\right)\log\left(1-p^{(j)}\right)\right) \\ &\text{subject to} \quad \frac{1}{U}\sum_{j=1}^{U} p^{(j)} = r \text{ and } p^{(j)} \in [0,1]. \end{aligned} \tag{69}$$

Similarly, this objective function can be optimized with a coordinate descent algorithm.

The gradient and Hessian matrix of the objective function are evaluated as follows:

$$\nabla_{\mathbf{w}} f(\mathbf{w};\mathbf{p}_t) = \lambda\mathbf{w} + \mathbf{X}^T\mathbf{v}, \tag{70}$$

$$\nabla^2_{\mathbf{w}} f(\mathbf{w};\mathbf{p}_t) = \lambda\mathbf{I} + \mathbf{X}^T(\mathbf{K}^T\mathbf{S}\mathbf{K}+\mathbf{Q})\mathbf{X}. \tag{71}$$

Here $\mathbf{v}$ is a vector with

$$\mathrm{v}_i = \mathrm{r}_i \alpha \ell_+^{(i)} (\ell_+^{(i)} - 1),$$

and r_i is defined as follows:

$$\mathrm{r}_i = \begin{cases} \frac{2(\mathrm{TP}+\mathrm{FP}+\mathrm{FN})}{(\mathrm{FP}+\mathrm{FN}+2\mathrm{TP})^2}, & \mathbf{x}^{(i)} \in \mathrm{X}_{\mathtt{pos}} \\ \frac{-2\mathrm{TP}}{(\mathrm{FP}+\mathrm{FN}+2\mathrm{TP})^2}, & \mathbf{x}^{(i)} \in \mathrm{X}_{\mathtt{neg}} \\ \gamma \frac{2(\mathrm{TP}+\mathrm{FP}+\mathrm{FN}) p_t^{(j)} - 2\mathrm{TP}(1-p_t^{(j)})}{(\mathrm{FP}+\mathrm{FN}+2\mathrm{TP})^2}, & \mathbf{x}^{(i)} \in \mathrm{X}_u \end{cases}$$

Here $\mathbf{S}$ is a symmetric matrix whose last column is defined in Table 10. The first two columns are the same as Table 7. Both $\mathbf{K}$ and $\mathbf{Q}$ are diagonal matrices defined as following:

$$\begin{aligned} \mathbf{K}_{ii} &= \alpha \ell_+^{(i)} (\ell_+^{(i)} - 1) \text{ and} \\ \mathbf{Q}_{ii} &= \mathrm{r}_i \alpha^2 \ell_+^{(i)} (\ell_+^{(i)} - 1)(2\ell_+^{(i)} - 1). \end{aligned}$$

Table 10: Last column of matrix $\mathbf{S}$

$\mathbf{S}_{ij}$	$\mathbf{x}^{(j)} \in \mathrm{X}_u$
$\mathbf{x}^{(i)} \in \mathrm{X}_{\mathtt{pos}}$	$\gamma \frac{4(\mathrm{TP}+\mathrm{FP}+\mathrm{FN}) p_t^{(j)} + 2(\mathrm{FP}+\mathrm{FN})(1-p_t^{(j)})}{(\mathrm{FP}+\mathrm{FN}+2\mathrm{TP})^3}$
$\mathbf{x}^{(i)} \in \mathrm{X}_{\mathtt{neg}}$	$\gamma \frac{2(\mathrm{FP}+\mathrm{FN}) p_t^{(j)} - 4\mathrm{TP}(1-p_t^{(j)})}{(\mathrm{FP}+\mathrm{FN}+2\mathrm{TP})^3}$
$\mathbf{x}^{(i)} \in \mathrm{X}_u$	$\gamma^2 \frac{4(\mathrm{TP}+\mathrm{FP}+\mathrm{FN}) p_t^{(i)} p_t^{(j)} + 2(\mathrm{FP}+\mathrm{FN}) p_t^{(i)} (1-p_t^{(j)}) + 2(\mathrm{FP}+\mathrm{FN}) p_t^{(j)} (1-p_t^{(i)}) - 4\mathrm{TP}(1-p_t^{(i)})(1-p_t^{(j)})}{(\mathrm{FP}+\mathrm{FN}+2\mathrm{TP})^3}$

With a fixed $\mathbf{w}_t$, the optimal label assignment vector $\mathbf{p}_{t+1}$can be obtained by optimizing the following constrained problem:

$$\begin{aligned} & \min_{\mathbf{p}} f(\mathbf{p}; \mathbf{w}_t) = -\frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \\ + \quad & \frac{T}{U} \sum_{\mathbf{x}_j \in X_u} \Big(p^{(j)} \log\left(p^{(j)}\right) + (1 - p^{(j)}) \log\left(1 - p^{(j)}\right) \Big) \\ & \text{subject to} \quad \frac{1}{U} \sum_{j=1}^{U} p^{(j)} = r \text{ and } p^{(j)} \in [0, 1]. \end{aligned} \tag{72}$$

Similarly, we can get the closed-form solution for this constrained optimization problem by Lagrange multipliers.

$$p_{t+1}^{(j)} = \frac{1}{1+\exp\left(\frac{\gamma U r^{(j)}-\nu}{T}\right)}, \tag{73}$$

$$\frac{1}{U}\sum_{\mathbf{x}_j \in X_u} \frac{1}{1+\exp\left(\frac{\gamma U r^{(j)}-\nu}{T}\right)} = r. \tag{74}$$

where $r^{(j)}$ is defined as following:

$$r^{(j)} = \frac{2(\mathrm{FP}+\mathrm{FN})(1-\ell_+^{(j)}) + 2\mathrm{TP}(1-2\ell_+^{(j)})}{(\mathrm{FP}+\mathrm{FN}+2\mathrm{TP})^2}.$$

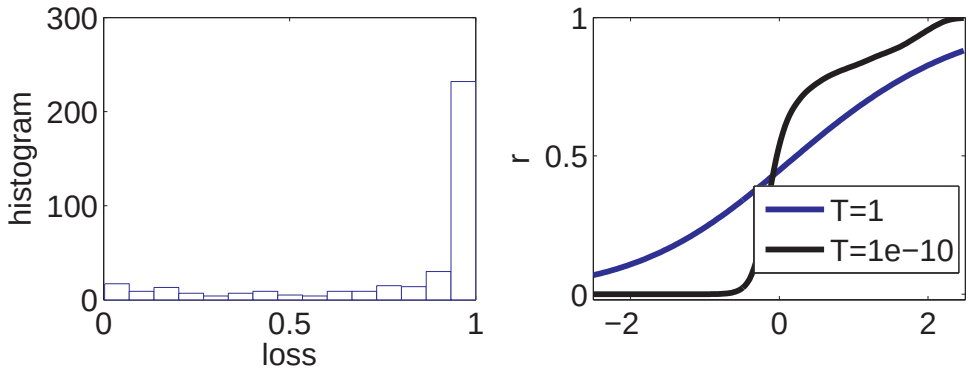

Figure 10: Nonlinear scalar equation for ν.

Figure 10 shows an example of the histogram of $\ell_+^{(i)}$ for some unlabeled data and the corresponding nonlinear equation at two different annealing temperature T.

3.3.4 Implementation Issues

To get the rMFoM criteria working properly and effectively in practical applications, there are several implementation issues that need to be accounted for carefully: 1) initialization of model parameters; 2) setting of nuisance parameters and 3) selection of optimization methods.

As mentioned in previous sections, the objective functions of rMFoM criteria are usually non-convex functions. So there is no guarantee of convergency to a global optimizer and the obtained local optimizer depends on the initial value of model

parameters. In the case of linear discriminant functions, there are three ways to initialize discriminant function parameters $\mathbf{w}$.

1. **Random initialization**: Random initialization is a simple initialization strategy and it works well in some practical applications.

2. **Initialization by Perceptron algorithm**: In this paper, the parameters of linear discriminant functions are initialized by Perceptron algorithm [23].

3. **Initialization by surrogate convex criteria**: Sometimes, it may be necessary to initialize the parameters of linear discriminant functions by optimizing a surrogate convex criterion. For instance, the model parameters can be obtained by optimizing the regularized logistic regression criterion, which is a convex function in terms of model parameters.

There are several parameters which have impacts on the rMFoM learning procedure in both supervised and semi-supervised learning scenarios.

As shown in Figure 8, α controls the slope of the sigmoid loss function. It's clear that when $\alpha = 20$, the sigmoid loss function is a very good approximation of the ideal 0-1 loss function used in binary classification. α also controls the number of instances involved in the model parameter update. When α increases, this number decreases. In this paper, α is set to 20 for all experiments, which has the same value as in [33] to make a fair comparison.

λ is a real-value regularization parameter in the rMFoM criteria. By varying λ, we can change the strength of the regularization term and get a regularization path of evaluation metrics at different λ. λ is introduced to improve the generalization capability within the Tikhonov regularization framework. From the perspective of numerical computation,λ is used to improve numerical stability, which is similar to the damping coefficient used in damped Newton method and Levenberg-Marquadt method to

overcome the singularity problem by adding a diagonal correction to the Hessian matrix [84]. If labeled data are sufficient, λ can be obtained by cross-validation. It's obvious that λ depends on the dimension of the feature vector. In our experiments, we found that $\lambda \in [10^{-7}, 10^{-3}]$ is a proper choice for text categorization with feature vector dimension larger than ten thousand.

γ is the parameter used in semi-supervised learning as a weighting factor for unlabeled instances. By varying the value of γ, we can control the contribution of unlabeled data to the final decision boundary. In this paper, γ is set to 1 for all experiments.

T is used in semi-supervised learning to control the deterministic annealing procedure. T is gradually decreased by multiplying a damping factor at each iteration [90] [95].

The proposed objective functions can be optimized efficiently. Batch gradient descent, conjugate gradient descent or stochastic gradient descent (a.k.a. generalized probabilistic descent [50]) can be applied to this problem. With the help of a proper line search scheme such as backtracking and Wolfe conditions [84], these first order optimization algorithms could converge to a local optimum. To speed up the convergence in large scale problems, a better choice is the quasi-Newton methods, e.g., limited-memory BFGS (L-BFGS) method [84], which uses the gradient vectors of previous iterations to approximate the Hessian matrix used in the current iteration and can yield superlinear convergence. In addition, the Hessian matrix doesn't need to be stored explicitly.

Trust region methods [16] [84] is another choice for many large scale optimization problems. It uses a local quadratic model to approximate the original objective function in the neighborhood of current parameters $\mathbf{w}_t$ as shown in following:

$$\min_{\|\mathbf{w}\|_2 \leq \Delta_t} m_t(\mathbf{w}) = f(\mathbf{w}_t) + \nabla f(\mathbf{w}_t)^T \mathbf{w} + \frac{1}{2} \mathbf{w}^T \mathbf{B}_t \mathbf{w}, \tag{75}$$

where Δ_t is the radius of trust region and $\mathbf{w}_{t+1}$ will be found by minimizing the

model function $m_t(\mathbf{w})$ within trust region. Here $\mathbf{B}_t$ could be the Hessian matrix, i.e., $\mathbf{B}_t = \nabla^2 f(\mathbf{w}_t)$ or BFGS approximated Hessian matrix. An advantage of trust region method is that it can work with non-symmetric positive definite Hessian matrices. A Newton trust region strategy allows the use of complete Hessian information even in regions where the Hessian has negative curvature. The trust region methods have a smooth transition from the steepest descent direction to the Newton direction. In the trust-region Newton Steihaug method [16], approximate conjugate gradient (CG) is used [16]. In each CG step, only a Hessian-vector multiplication is needed, which means that we don't need to compute and store the Hessian matrix explicitly. So we can leverage the sparsity pattern of the Hessian matrix efficiently.

3.3.5 Experimental Study for Supervised Learning

Several experiments were conducted on a large scale text categorization dataset to demonstrate the effectiveness of the regularized MFoM learning criteria for supervised learning tasks. The same dataset was used in the original MFoM learning paper [33].

In TC experiments, a document is usually represented by a vector in a high-dimension vector space using a bag-of-words (BoW) model (or uni-gram model), where each dimension of the vector corresponds to an element of a large lexicon. An advantage of this representation is that variable length of documents can be converted into fixed length vectors. For many real-world TC tasks, the size of the lexicon is usually greater than ten thousand. In this paper, the ModApte version of the Reuters-21578 corpus is used to evalute the supervised rMFoM criteria, which is a standard benchmark dataset for TC evaluation. To increase the discrimination power of such representation, some preprocessing and normalization have been conducted as described in [33] [32]. The size of our lexicon is 10,118 and there are 90 categories preserved in the corpus after preprocessing. The training set consists of 7,700 documents and testing set has 3,019 documents.

To reduce the dimensionality of each document vector, latent semantic analysis (LSA) is usually used to project the original vectors into a compact latent semantic space. The original MFoM paper showed that using a 1,613 dimensional LSA vector can achieve the same performance as the original 10,118 dimensional vector [33]. A drawback of such a compact representation is that the sparsity of the original term-document matrix is lost because the LSA term-document matrix is usually a dense matrix. For instance, the original term-document matrix (10,118 * 7,770) of training set has only 381,249 non-zero elements, i.e., 49 non-zero elements per document. After SVD factorization, the new term-document matrix (1,613 * 7,770) in the latent semantic space is about 30 times denser with 12,533,010 non-zero elements. It's more computational and memory consuming because it require more memory space and more float computation. So in this paper, we are working in the original vector space.

The performance of TC is evaluated in terms of precision, recall and F1 measure for each category as defined in Table 6. The average performance over all 90 categories is indicated by the macro-average F1 ($\mathrm{F1}^M$) and micro-average F1 ($\mathrm{F1}^\mu$) defined as following:

$$\mathrm{F1}^M = \frac{2\sum_{i=1}^{K}\mathrm{R}_i\sum_{i=1}^{K}\mathrm{P}_i}{K(\sum_{i=1}^{K}\mathrm{R}_i+\sum_{i=1}^{K}\mathrm{P}_i)} \tag{76}$$

$$\mathrm{F1}^\mu = \frac{2\sum_{i=1}^{K}\mathrm{TP}_i}{2\sum_{i=1}^{K}\mathrm{TP}_i+\sum_{i=1}^{K}\mathrm{FP}_i+\sum_{i=1}^{K}\mathrm{FN}_i} \tag{77}$$

So in this section, we use the rMFoM criteria to directly maximize F1 measure on the training set as reported in [33].

Previous studies of MFoM learning algorithms for text categorization on Reuters-21578 obtained slightly better result than SVM classifiers [33]. In our experiments, we compare the original MFoM and the regularized MFoM on Reuters-21578 to investigate the importance of regularization.

Table 11 summarizes the results for top-10 topics that have more than 180 positive training samples for each topic. For the top-10 topics, regularized MFoM gives much

Table 11: Performance comparison for top-10 topics.

Category	MFoM	rMFoM
Earn	0.979	0.987
Acq	0.961	0.967
Money-fx	0.769	0.836
Grain	0.931	0.949
Crude	0.872	0.904
Trade	0.746	0.808
Interest	0.769	0.802
Wheat	0.832	0.892
Ship	0.842	0.884
Corn	0.850	0.867
Micro-avg(all 90)	0.869	0.887
Macro-avg(all 90)	0.531	0.561

better results than the original MFoM criteria with linear classifiers. The performance improvement is statistically significant and the p-value of the Wilcox paired signed rank test is 0.002. We can conclude that regularization is very effective for topics having larger number of positive samples.

As discussed in previous sections, λ has great importance on system performance. We investigated the system performance with different λ for the topic "earn" in Reuters-21578 dataset. The right part of Figure 11 shows that $\mathbf{w}$ decreases when λ increases. The left part shows the F1 measure on both the training and test set at different λ. It shows that a proper selection of regularization terms will improve the generalization capability. The λ will also improve the stability of numerical computation.

We now investigate some properties of rMFoM criteria by looking into the class conditional distributions $p(\mathbf{w}^T\mathbf{x}|y)$ obtained by different criteria as shown in Figure 12. It is clear that both MFoM and rMFoM could improve the separation between positive and negative classes over the original random initialized models. The original MFoM learning tends to have larger values for model parameters so that both distributions have longer support and tails. The regularized MFoM classifiers have a

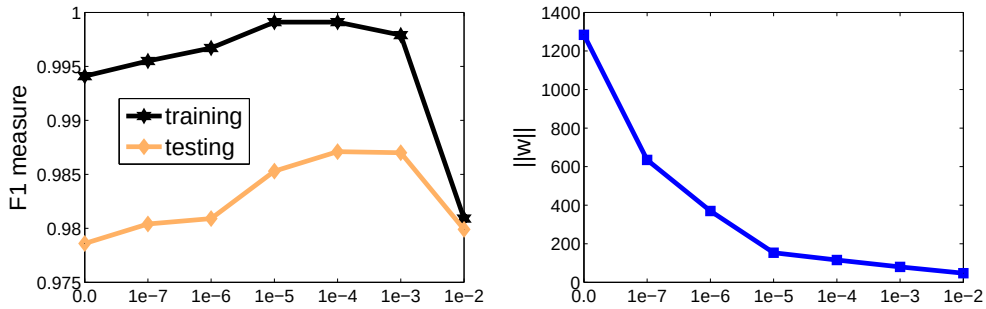

Figure 11: Regularization path in rMFoM learning.

more compact distribution while keeping good separation.

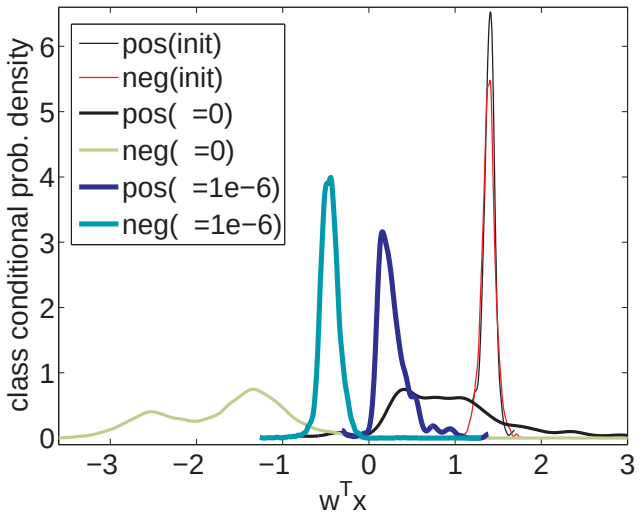

Figure 12: Class conditional probability density of $\mathbf{w}^T\mathbf{x}$.

3.3.6 Experimental Study for Semi-supervised Learning

We also conducted text categorization experiments to demonstrate the effectiveness of rMFoM on semi-supervised learning.

For semi-supervised rMFoM learning, two widely used datasets aut-avn and real-sim are employed to evaluate the proposed semi-supervised classification algorithms [95]. These two datasets come from a collection of UseNet articles from four discussion groups, for simulated auto racing, simulated aviation, real autos, and real aviation. The basic information about these two datasets are listed in Table 12, where the

dimension of vectors, number of training and testing instances, and the ratio of positive instances in the whole dataset are listed. To get a reliable evaluation, these two datasets are randomly split into 10-fold training data (both labeled and unlabeled) and test sets. Because the data from positive and negative classes are fairly balanced, we use the rMFoM criteria to directly optimize the classification error on the training data as in some earlier work [95] [13].

Table 12: Two semi-supervised learning datasets

Dataset	dim.	train (l+u)	test	r
aut-avn	20707	35588	35587	0.65
real-sim	20958	36155	36154	0.31

We first investigate the effectiveness of rMFoM criteria for semi-supervised learning tasks.

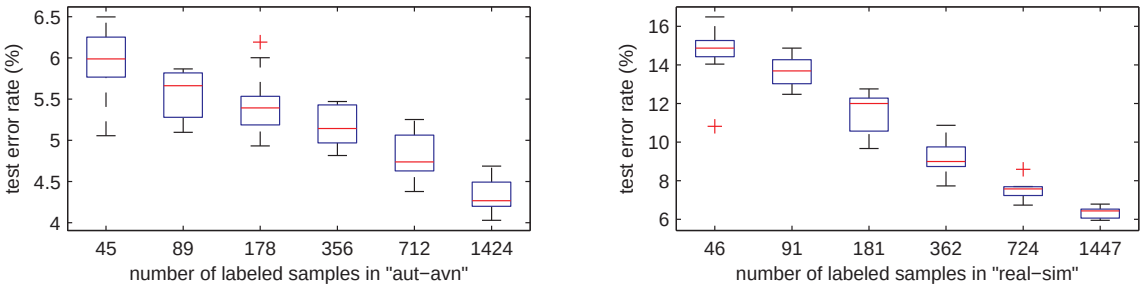

Figure 13: Test error rate of semi-supervised rMFoM learning on two datasets.

Figure 13 shows the boxplot of the test error rate of semi-supervised rMFoM learning as a function of the number of labeled instances on two datasets, which clearly shows the median and variance of test error rate at each setting. It's obvious that with additional labeled data become available, the system performance will be improved. However, when the proportion of labeled data reach some ratios, e.g. 5%, the system performance tends to be saturated. It means that for some real-world applications, it's possible to label only a small part of training data while achieving very good performance.

Table 13: Comparison with supervised learning.

aut-avn	**l=45**	**89**	**178**	**356**	**712**	**1424**
SVM	31.8	24.0	15.6	10.3	7.6	5.8
rMFoM	5.9	5.6	5.5	5.2	4.8	4.3
real-sim	**l=46**	**91**	**181**	**362**	**724**	**1447**
SVM	28.7	24.9	18.2	12.8	9.8	7.5
rMFoM	14.6	13.7	11.5	9.2	7.5	6.4

In Table 13, we compare the semi-supervised rMFoM learning with SVM that only use the labeled instances for training. It is clear that with rMFoM criteria, the unlabeled data can improve the system performance significantly.

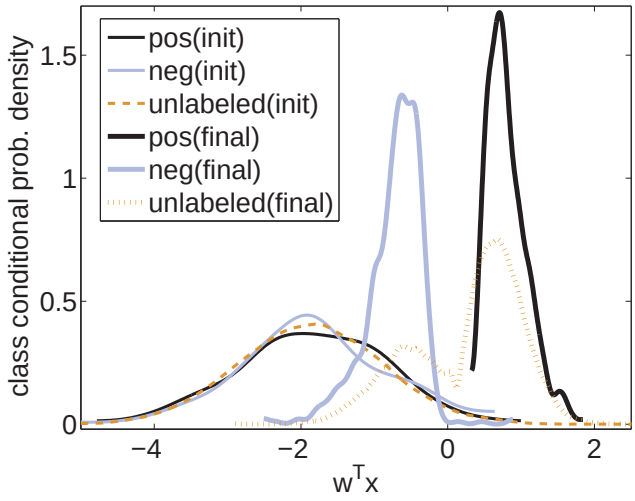

Figure 14: Class conditional probability density of $\mathbf{w}^T\mathbf{x}$.

In Figure 14, we demonstrate the class conditional probability density of the misclassification measure in semi-supervised learning. In the beginning, $\mathbf{w}^T\mathbf{x}$ for positive class, negative class and unlabeled data overlapped each other. After the rMFoM learning, positive and negative instances are well separated. Meanwhile, unlabeled training data demonstrate a clear two-modal distribution which correspond to positive and negative classes. And it explains why unlabeled data will help to improve system performance in semi-supervised learning tasks.

3.3.7 Discussions

We present a regularized maximum figure-of-merit (rMFoM) approach for supervised and semi-supervised learning, and demonstrate the effectiveness of the rMFoM criteria and the importance of regularization by several large-scale text categorization experiments.

The regularized MFoM criteria have all the advantages of the original MFoM learning criteria: (1) explicitly optimize the metrics of interest to keep the consistency between model training and performance evaluation. It can be used for any discriminant function, including both generative models and discriminative models; (2) The performance measures obtained in the training stage can also be used to predict the performance on a similar testing set. This is highly beneficial when it is expensive to have a separate evaluation set; (3) It is robust and still effective when only a small number of positive instances are available. It is also as insensitive to the training data variation and unseen testing conditions.

By reformulating the MFoM learning criteria in the Tikhonov regularization framework, the rMFoM criteria seeks to improve the generalization capability of any classifiers based on discriminant functions.

Another important novelty of this study is that the rMFoM learning criteria is successfully extended to more general semi-supervised learning scenarios. It provides us a theoretical justification on the widely used self-training algorithm.

The proposed rMFoM learning approach with LDF has very good scalability. As shown in our experiments on large scale text categorization, it can handle datasets with huge number of instances and feature dimensions. The optimization techniques discussed in this paper have less memory and computational demand.

We conducted comprehensive experimental studies of rMFoM criteria on several large scale text categorization tasks. We compared the MFoM-learned classifiers with the binary tree classifiers learned by MFoM criteria, our results showed that the

F1-based rMFoM is able to achieve high performance by introducing a properly set regularization term.

CHAPTER IV

A DETECTION-BASED ASR SYSTEM

The state-of-the-art hidden Markov model (HMM)-based automatic speech recognition (ASR) systems [87] are generally formulated as follows:

$$\begin{aligned} \hat{W} &= \arg\max_{W_1^n} p(O_1^T|W_1^n)P(W_1^n) \\ &= \arg\max_{W_1^n} \left[\sum_{S_1^T} p(O_1^T|S_1^T)p(S_1^T|W_1^n)\right] P(W_1^n), \end{aligned} \tag{78}$$

where W_1^n is a sequence of n words, O_1^T is a sequence of T acoustic observations, S_1^T is any state sequence of length T, $P(W_1^n)$ is the probability of a particular word sequence W_1^n obtained from a language model (LM), and $p(O_1^T|W_1^n)$ is the acoustic model (AM) representing the probability of an observation sequence O_1^T given the word sequence W_1^n. This is the well-known maximum *a posterior* (MAP) channel decoding framework for ASR aiming at finding the word sequence $\hat{W}$ with maximum likelihood for a given observation sequence O_1^T.

All the acoustic knowledge (i.e., state probability distributions), phonetic knowledge (i.e., context dependency and phonological rules), lexical knowledge (i.e., pronunciation dictionary), linguistic knowledge (i.e., grammar and language models), and practical constraints are integrated into a huge decoding network for recognition [44], which is a typical expectation-driven top-down processing approach.

The HMM-based ASR systems have witnessed dramatic progress and great success in the last several decades. More improvements have been obtained in the field of speech and language modeling due to the extensive use of statistical learning techniques and more speech and language data collections. One possible reason why HMMs are used in speech recognition is that a speech signal could be viewed as a

short-time stationary signal. Speech could thus be modeled with a Markov model. Another reason why HMMs are so popular is because their parameters can be estimated automatically and efficiently, and HMMs are simple and computationally feasible in practical applications.

However, ASR systems performs much worse than human speech recognition (HSR) in most situations. The variability in speakers, acoustic channels, and environment noises will greatly degrade the ASR performance, because it is impossible or very expensive to collect sufficient data to cover all the uncertainties in real-world situations. To partially compensate the acoustic and linguistic mismatch, some attempts were made to find robust distinctive features which are invariant to speakers and speaking environments [67] [49]. In addition, some adaptation techniques have been developed. The basic idea is to compensate for the mismatch between training and test conditions by transforming the signal, feature vectors, or model parameters based on adaptation data that are collected for task-specific applications. For instance, acoustic model adaptation using maximum likelihood linear regression (MLLR) [61], maximum a posterior (MAP) criterion [35]), and language model adaptation with maximum entropy (ME) approach [91] have been widely used.

In some applications, such as broadcast news transcription or human-machine dialog systems, adaptation techniques are still insufficient. There are many out of vocabulary (OOV) words, out of grammar sentences, and many other unexpected audio events. For instance, in human-machine spoken dialog systems, all the expected and acceptable sentences are represented by a deterministic finite state grammar (FSG). Nevertheless, a wide sentence variation including extraneous words, hesitations, repetitions, disfluency, and many other unexpected expressions were observed in practical scenarios [51].

Some pioneering studies on a key-phrase detection-based spoken dialog system have been proposed [51], where a group of task-specific key-phrases are first detected

and then the verifiers at the phrase and sentence levels are used for decision making.

4.1 Overview of Detection-based ASR Systems

Inspired by the studies of cognitive psychology and linguistics, a detection-based framework has been proposed as an alternative framework for speech recognition recently [59] [58]. In cognitive psychology, human speech recognition could be a mixture of bottom-up and top-down processing. In linguistics, it is features and not phonemes that are viewed as the fundamental units of speech, where phonemes are coded in terms of distinctive features. The brain recognizes sounds by doing a distinctive feature analysis from the information going to the brain. These features are somewhat insensitive to speaker, noise, background, and reverberation, i.e., they are robust and reliable for auditory perception and speech recognition. Therefore, some research efforts have been focused on the detection of low level distinctive attributes and knowledge integration schemes in detection-based speech recognition systems. The detection-based framework approaches ASR from a more linguistic perspective. Speech is modeled as connected sequences of interacting features rather than individual phoneme segments. One key feature of the detection-based approach is that the outputs of the detectors do not have to be synchronized in time and therefore the system is flexible enough to allow a direct integration of both short-term detectors, e.g., for detection of voice onset time (VOT), and long-term detectors, e.g., for detection of pitch, syllables, and particular word sequences.

What we intend to do is to simulate the human auditory perception to some extent. We want to incorporate some low level acoustic-phonetic features into the standard HMM-based ASR system to bridge the gap between ASR and HSR. Meanwhile, we could improve its robustness and accuracy.

Some knowledge supplemental systems have been developed, where the evidence from the low level attributes (such as the manner and place attributes) was combined

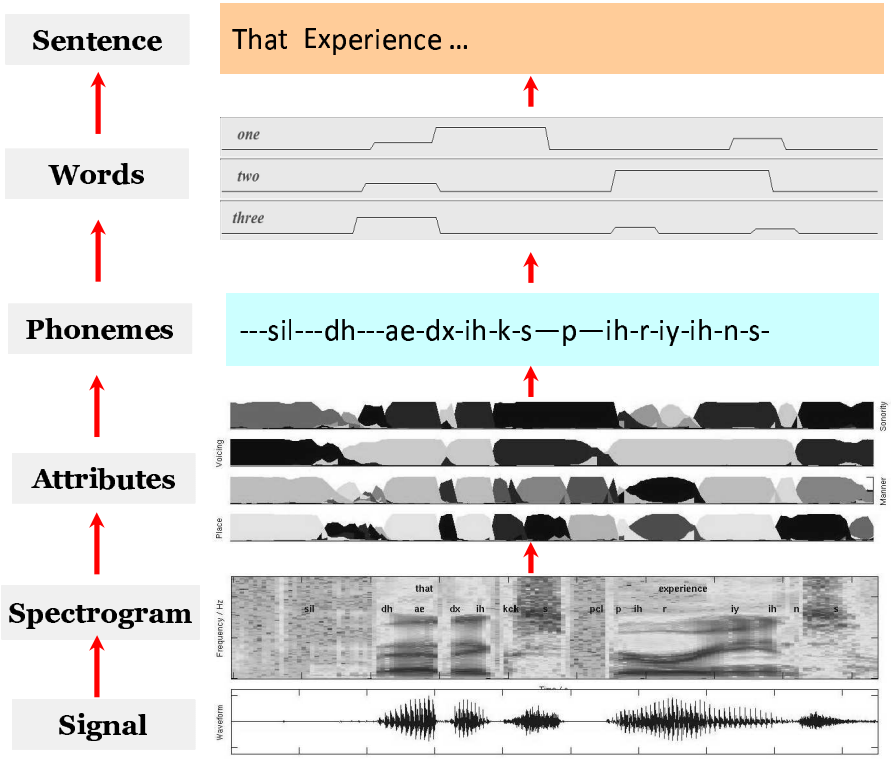

Figure 15: Bottom-up knowledge hierarchy for ASR.

with the N-best list [63] or the lattice [96] generated by the conventional HMM based ASR system. However, these knowledge supplemental techniques are still insufficient to incorporate all the knowledge sources into a single search network as required by the maximum a posterior (MAP) decoding paradigm. So in this study, a bottom-up detection-based framework is proposed to allow more flexibilities in knowledge integration.

Our detection-based ASR system has several components: (1) individual detector design; (2) fusion algorithms in the attribute detection hierarchy; (3) integration with the standard HMM-based system. The key components of our system are a collection of articulatory feature detectors and most of these detectors are language independent.

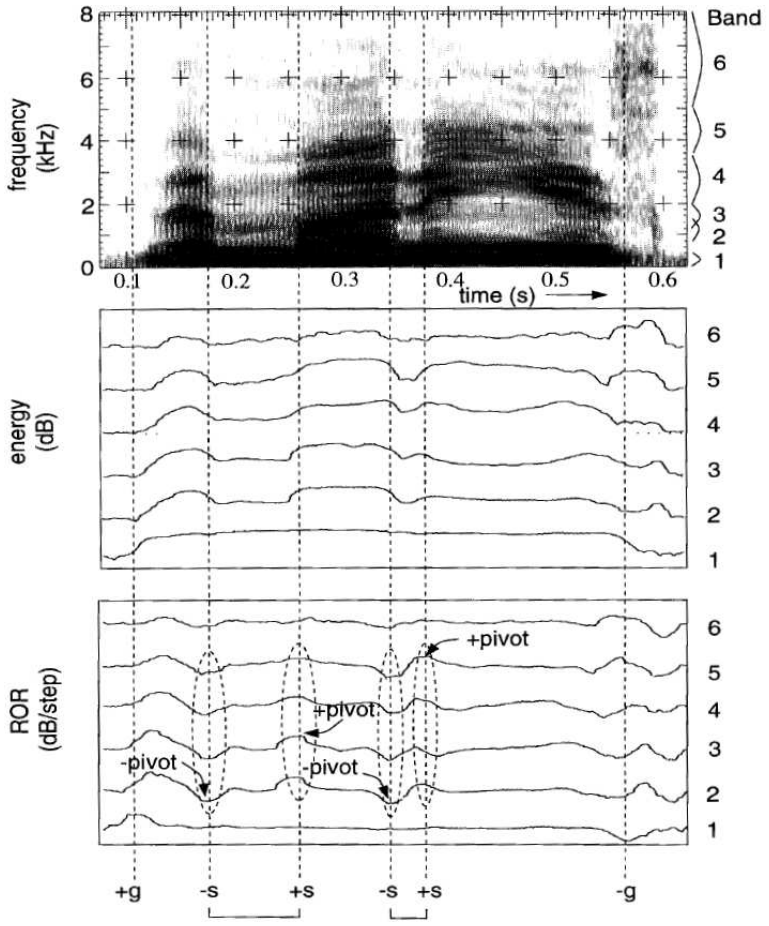

Figure 16: Energy-based landmark detection (from [67]).

4.2 Landmark Detection

Landmarks are times in an utterance when the acoustic correlation of distinctive features are most salient. In a pure knowledge-based speech recognition system, landmarks are elementary components. In this study, we follow the exact signal processing methods described in [67] for landmark detection. As shown in Figure 16, energy contours in 6 frequency bands are computed first and then pivot points are detected. Based on phonetic and phonological knowledge, some of those pivots are verified and classified as one of 4 landmarks investigated in our study, which are summarized as follows.

- G(lottis): [+g] means turning on of glottal vibration and [-g] means turning off.

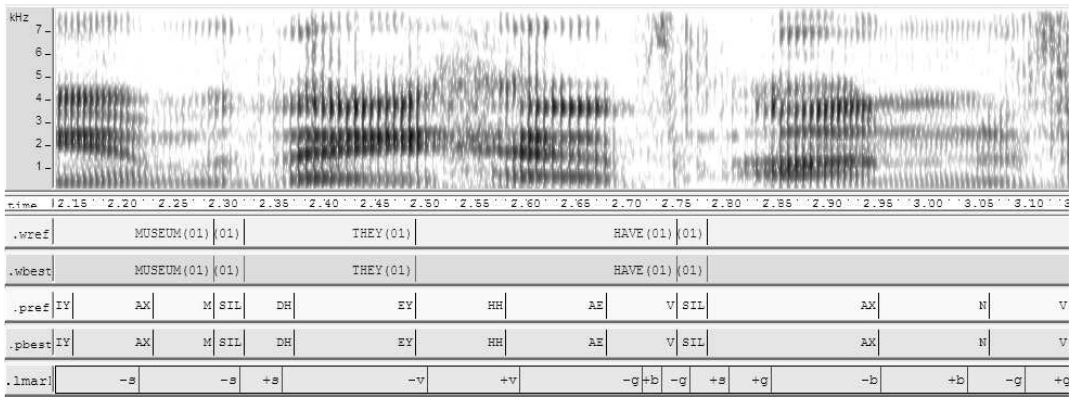

Figure 17: Landmark detection example.

- S(onorant): [+s] means end of sonorant and [-s] means begin of sonorant.
- B(urst): [+b] means burst start and [-b] means burst end.
- V landmarks are missed "g" landmarks but detected based on higher band rate of rise (ROR) peaks.

Figure 17 shows an example of the detected landmarks for a speech segment. Some intuitive interpretation about these landmarks are as follows: a [-s] landmark shows the start of a sonorant (nasal + semi-vowel) sound and the region between a [+g] and [-g] landmark is a voiced sound. [+b] indicates the start of a burst. The performance of landmark detection on a variety of experiment conditions and datasets was described and summarized in [67].

4.3 Speech Attribute Detection

As stated above, from a linguistic perspective, it is distinctive features not phonemes that are the basic units of speech recognition. In this study, these distinctive features will be part of our speech attribute collection.

The following are some of the speech attributes that will help improve speech recognition: (1) manner of production (oral, nasal, fricative, or involving a partial blockage of the airflow); (2) place of articulation (dental, lip, etc.); (3) voicing: the

larynx vibrates for a voiced phoneme and does not for a voiceless phoneme. (4) prosodic features of speech signals include duration, pitch, stress, and loudness.

4.3.1 Manner of Production

In linguistics (articulatory phonetics), manner of articulation describes how the tongue, lips, jaw, and other speech organs that are involved in making a sound make contact. This concept is usually only used for the production of consonants. In English, consonants that have the same place of articulation are said to be homorganic. For any place of articulation, there may be several manners, and therefore several homorganic consonants. Usually, the following manners of articulation are studied for English sounds.

Stop Also named plosive, it is a complete occlusion of both the oral and nasal cavities of the vocal tract, and therefore no air flow occurs. Stop sounds in English consists of /p/, /t/, /k/ (unvoiced) and /b/, /d/, /g/ (voiced).

Nasal There is complete occlusion of the oral cavity, and instead the air passes through the nose. The shape and position of the tongue determine the resonant cavity that gives different nasal stops their characteristic sounds. Nasal sounds in English consists of /m/, /n/ and /ng/.

Fricative There is continuous frication (turbulent and noisy airflow) at the place of articulation. Fricative sounds in English consists of /f/, /s/ (unvoiced) and /v/, /z/ (voiced), etc.

Affricate It begins like a stop sound, but it releases into a fricative rather than having a separate release of its own. Affricate sounds in English consists of /ch/ and /j/.

Glide It pronounced like a vowel but with the tongue closer to the roof of the mouth, so that there is a little obstruction and slight turbulence. In English, /w/ is the

semivowel equivalent of the vowel /u/, and /y/ is the semivowel equivalent of the vowel /i/ in this usage. /r/ and /l/ are also glides.

4.3.2 Place of Articulation

In articulatory phonetics, the place of articulation of a consonant is the point of contact, where an obstruction occurs in the vocal tract between an active (moving) articulator (typically some part of the tongue) and a passive (stationary) articulator (typically some part of the roof of the mouth). Along with the manner of articulation and phonation, this gives the consonant its distinctive sound.

A place of articulation is defined as both the active and passive articulators. For instance, the active lower lip may contact either a passive upper lip (like [m]) or the upper teeth (like [f]). There are five basic active articulators: the lip ("labial consonants"), the flexible front of the tongue ("coronal consonants"), the middle/back of the tongue ("dorsal consonants"), the root of the tongue together with the epiglottis ("radical consonants"), and the larynx ("laryngeal consonants"). These articulators can act independently of each other, and two or more may work together in what is called coarticulation.

The passive articulation, on the other hand, is a continuum without many clear-cut boundaries. Some places of articulation such as palatal and velar merge into one another, and a consonant may be pronounced somewhere between the named places.

4.3.3 Segment-based Attribute Detection using HMMs

The manner and place of articulation can be detected by either segment-based or frame-based data-driven methods. For segment-based attribute detection, each manner and place of articulation is modeled by two HMMs: one target model (e.g., vowel) and one imposter model (e.g. non-vowel) or a collection of competing cohort models. It is essentially a hypothesis verification problem and log-likelihood ratio (LLR) or generalized log-likelihood ratio (GLLR) is used as the detection statistics [70].

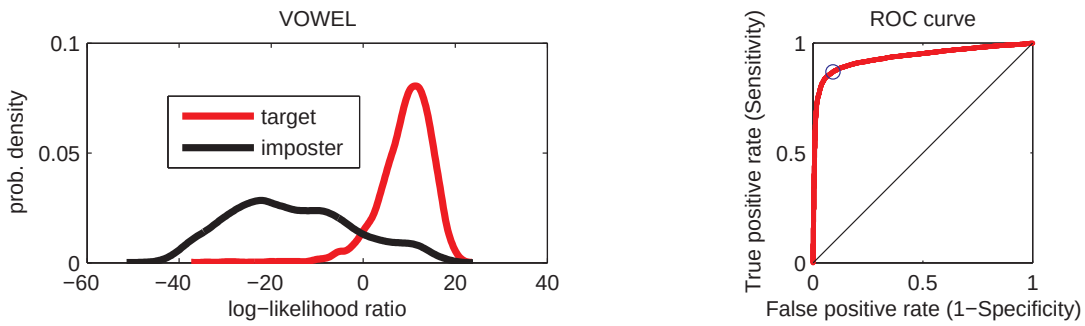

Figure 18: Attribute verification for VOWEL on WSJ.

Performance evaluation experiments have been conducted for 15 acoustic-phonetic attributes on both the WSJ nov92 evaluation dataset (330 utterances) and the RT03 dataset (468 segments). All the utterances were aligned using a set of cross-word triphone HMMs and the segment boundary information is obtained from the alignment. All the target and imposter HMMs are trained on the TIMIT dataset to simulate the acoustic mismatch conditions. On the WSJ dataset, there are 23,593 segments being evaluated with respect to 15 pairs of models, and on the RT03 dataset, there are 95,449 segments obtained from alignment.

From our experiments, the detection performance is good in terms of ROC and equal error rate (EER) for some attributes, such as vowel shown in Figure 18 and Figure 19, while the detection performance for some attributes is poor, such as labial shown in Figure 20 and Figure 21. We conclude that the place of articulation is much harder to detect than the manner of articulation. It would be better to consider the context of each attribute for the place of articulation.

4.3.4 Frame-wise Attribute Detection using ANNs and SVMs

Two frame-wise attribute detection approaches were investigated in our study without the phoneme boundary information: artificial neural networks (ANNs) and support vector machines (SVMs).

A multiple layer perceptron (MLP) neural network is trained for each manner

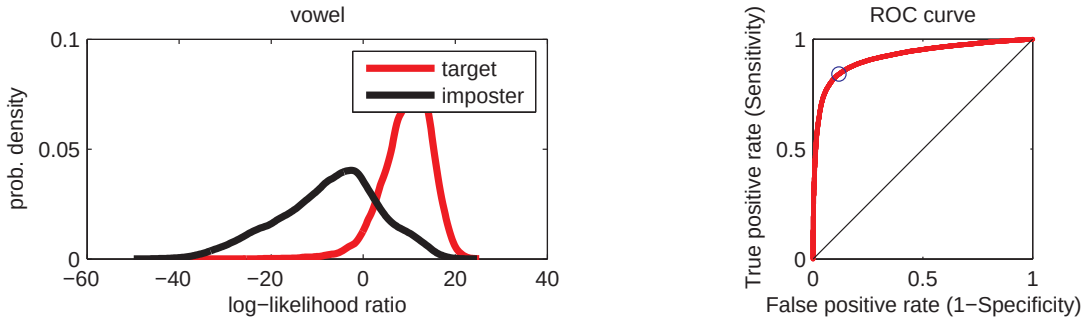

Figure 19: Attribute verification for VOWEL on RT03.

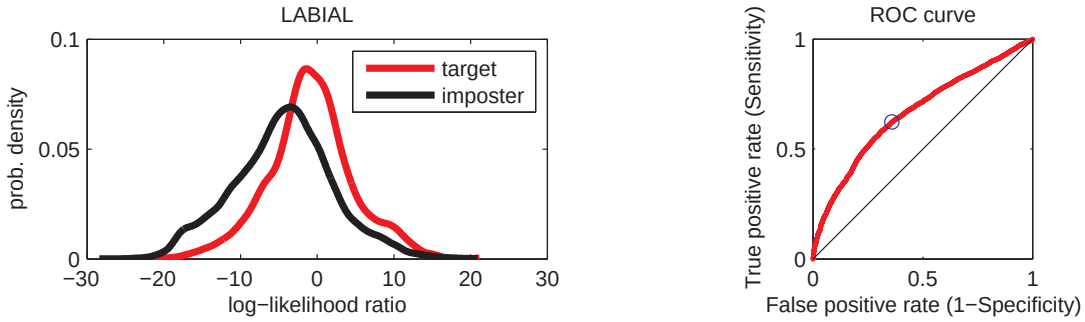

Figure 20: Attribute verification for LABIAL on WSJ.

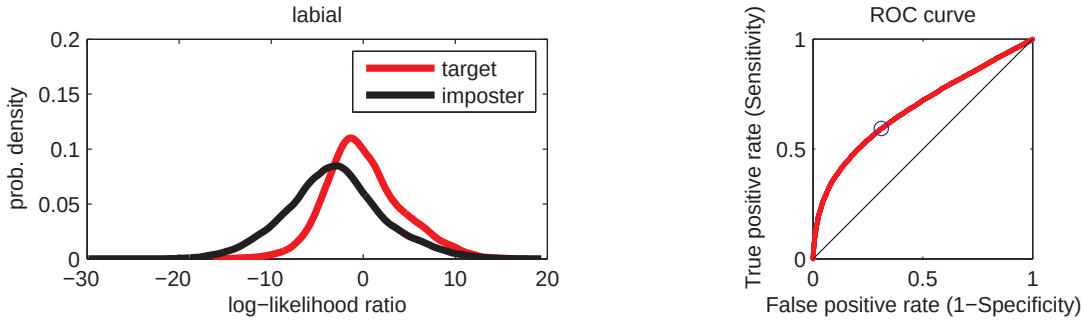

Figure 21: Attribute verification for LABIAL on RT03.

and place of articulation and its output can be interpreted as an approximation of posterior probability or confidence measure of detection. A MLP is trained on a set of 3-hour broadcast news speech (2.5-hour for training and 0.5-hour for cross-validation). The raw feature used as input of MLP is the TRAP feature, which is the temporal trajectory of log energy from 31 consecutive frames (± 15 frames of a central frame) for each critical band. Band-wise mean and variance normalization was also conducted. Then discrete cosine transformation (DCT) is applied to each band to perform dimension reduction and de-correlation. Only the first 10 DCT coefficients and the log-energy of each band are preserved in our experiments. There are 23 critical bands used for 16k Hz speech data. TRAP features were extracted using the trapper software[1].

The topology of the MLP has 253 input units, 800 hidden units and 2 output units that corresponds to target attribute and its competing model. The MLP was trained using the ICSI's QuickNet toolkit[2]. For SVM based detectors, both PLP and FMPE features were investigated in our experiments. Table 14 shows a comparative study of frame-wise ANN and SVM attribute detectors.

As expected, the detection results are quite noisy compared to segment-based detectors. We also conclude that place of articulation is much harder to detect than manner of articulation, which is consistent with the segment-based detectors.

4.4 Construction of Knowledge Hierarchy

After the lower level speech attributes have been detected, we could construct the task-specific knowledge hierarchy level by level. In chapter 3, we have described several approaches for information integration. In this section, we will demonstrate how to combine manner and place of articulation to phoneme level through a phoneme classification experiment.

[1]http://speech.fit.vutbr.cz/files/software/trapper.html
[2]http://www.icsi.berkeley.edu/Speech/qn.html

Table 14: Comparison of frame-wise ANN and SVM detectors.

	ANNs			SVMs	
Attribute	Train	CV	Test	PLP	FMPE
back vowel	88.64%	86.68%	86.68%	85.23%	87.67%
continuant	86.15%	82.77%	82.50%	75.76%	83.00%
coronal	85.94%	82.29%	82.70%	76.25%	84.11%
fricative	92.07%	88.86%	89.04%	85.84%	87.70%
high vowel	88.86%	86.46%	86.24%	84.90%	88.86%
labial	90.47%	88.26%	88.03%	86.11%	89.04%
low vowel	91.86%	89.21%	89.38%	90.37%	92.41%
nasal	92.96%	90.76%	90.63%	88.69%	91.35%
retroflex	94.74%	93.15%	93.07%	93.95%	94.86%
round	90.92%	88.19%	88.48%	86.67%	89.89%
semivowel	91.39%	89.75%	89.54%	87.44%	90.34%
sonorant	87.42%	84.61%	83.81%	81.33%	85.02%
stop	89.11%	86.40%	86.56%	81.56%	84.68%
vowel	86.39%	83.68%	83.35%	79.10%	85.13%

The maximum entropy (MaxEnt) model was used as the fusion strategy. Input to the MaxEnt model is the detected landmarks and articulatory attributes, which were represented by some feature functions $f_k(x, y)$ as follows:

$$f_{/n/,\text{nasal}}(x, y) = \begin{cases} 1, & \text{if } y = /n/ \text{ and nasal}(x) = \text{true} \\ 0, & \text{otherwise} \end{cases}$$

A feature function is nonzero only if the label y match the label of the observation, which is used to indicate the existence or nonexistence of a particular attribute, landmark or phone class in the observed speech segment. For example, the above feature function indicates that a particular phone label (/n/) is dependent on the existence or nonexistence of nasality in the observed data in a segment. We create a feature function for each label/attribute pair. In addition, a bias feature function is nonzero if the label that they are defined for occurs.

Phoneme classification experiments were conducted on the TIMIT dataset and we follow the same experiment settings as described in [37]. The output of each ANN detector was discretized and then employed in feature functions. Table 15 shows the

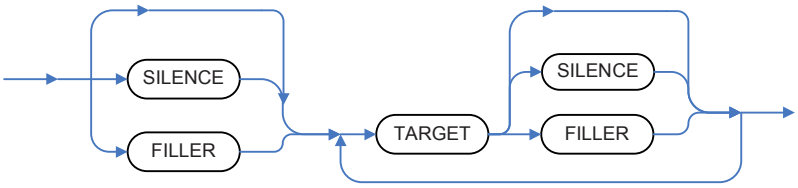

Figure 22: A single word detector.

phoneme classification results of three systems. We conclude that MaxEnt fusion could achieve comparable or slightly better results than HMM-based system using MFCC features, while it is worse than the results using hidden conditional random field (HCRF) [37].

Table 15: Phoneme classification error rate.

MaxEnt	HMM(MFCC+ML)	HCRF (L-BFGS)
27.3%	28.1%	21.7%

4.5 Information Integration by Hard Decision

Instead of using the low level attributes to construct a knowledge hierarchy, we can incorporate these information into either the word-level detectors by hard decision or the state-of-the-art large vocabulary continuous speech recognition (LVCSR) system by soft decision.

This section demonstrates an implementation of a single word detector and digits recognition. The system consists of three parts: (1) word detector design; (2) knowledge guided word hypotheses verification and false alarm pruning; (3) combining word hypotheses into word strings. The system is realized for connected digit recognition task and selected function words and content words recognition on a LVCSR task. Figure 22 is a prototype of a detection-based digit recognition system.

In a detection-based ASR framework, each lexical item in the vocabulary has a separate detector. In this implementation, HMM modeling techniques are used for single word detector design. One of the crucial issues is to choose an appropriate

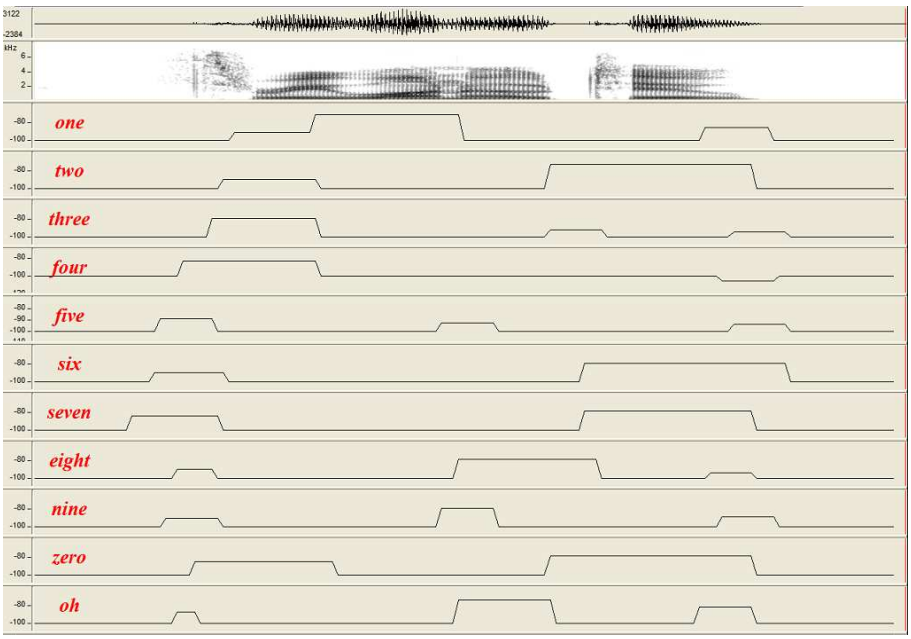

Figure 23: Word candidates from 11 detectors.

grammar network [74] [71].

Figure 22 shows the grammar network used in this study. For each target word, a collection of filler (or cohort) models and a silence model will compete with it. According to the basic requirement of candidate detection, fewer misses are expected. For each target word, we can choose several competing words as its cohorts. More generally, a phone-loop network is used as the filler model to absorb all the other events except for the target word. Using this network, the target word can be detected with a very high detection probability.

Figure 23 shows an example of the output of 11 digit detectors. The first and second panels are the waveform of the test utterance 31o2 and its corresponding spectrogram, respectively. The following 11 panels are the detector outputs, with the level on the y-axis for each panel indicating a confidence measure for detecting these words. For example, the bottom panel has three segments above the x-axis. It means that the "oh" detector tells us these segments are digit "oh." Actually, only the second segment is really a digit "oh." The first and third ones are false alarms.

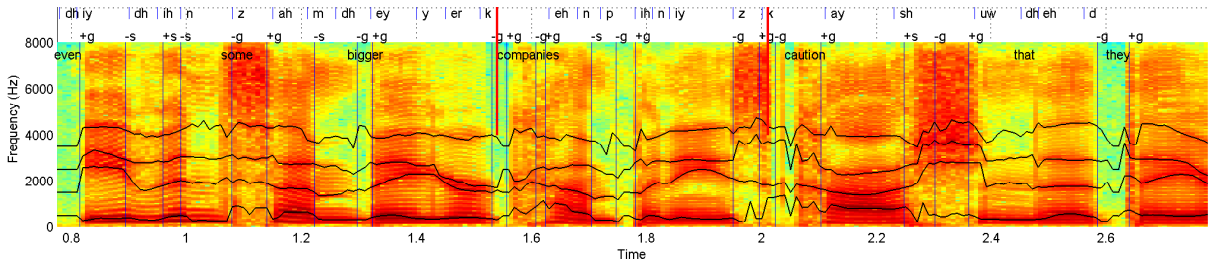

Figure 24: Knowledge sources for detected words.

Figure 24 shows the detected knowledge sources using phoneme detectors, landmark detectors, and word detectors for a LVCSR task [71]. The top-most panel is the recognized phone sequence using an ANN-based phone recognizer. The second sequence is landmarks, as described in [67]. The speech segment between the two arrows is a hypothesized segment for the word "company." The vertical lines indicate the location of each landmark. For example, a [-s] landmark shows the start of a nasal sound. The region between a [+g] and [-g] landmark is a voiced sound. For this voiced region, we can further analyze the formant transition pattern. A similar procedure can be applied to the voiceless region indicated by [-g] and [+g] landmarks.

4.5.1 Word Pruning and Verification

After the word candidates have been detected, a word level information fusion and verification were conducted. Three pruning strategies are used to incorporate attribute information by hard decision.

First, phoneme-dependent duration constraint is a simple pruning strategy. The duration constraints can be used to eliminate those very short segments from the detection results. The statistics of phoneme duration were obtained from the aligned training set. For example, the duration of word "one" (/w/-/ah/-/n/) should be greater than 150 ms.

The second method is to use the models of the manner and place attributes to generate the attribute sequence for each detected segment. Each manner attribute

is modeled with a HMM. Then for each detected segment, it can be decoded as a sequence of manner attributes. If correctly decoded, each word has its own attribute sequence pattern. Any obvious deviation from the desired pattern can be pruned by some rules. For example, among all the outputs of detector "one", some of them are actually from speech for "nine". So we can prune those segments whose manner attribute sequence doesn't contain glides. This kind of model based pruning techniques have shown their effectiveness in our evaluation experiments.

Model based pruning can easily be implemented and used. However, we still need to train these manner attribute models from some training set. Inevitably, the robustness problem still exists. So it's desirable to have some robust pruning strategies. Signal feature based pruning is one of them. For example, from research in acoustics, we know that the energy of a nasal sound /n/ is often concentrated on the low frequency region (below 400 HZ), while the fricative sound /f/ has a relatively flat spectrum and energy distribution in high frequency region. So the percentage of low frequency energy in the total energy is useful and robust in distinguishing the nasal and fricative sound. Also the formants position of vowels and other spectral features can be used to distinguish certain pair of sounds.

4.5.2 Hypothesis Combination

A weighted directed graph (WDG) is one of the methods that can be used to combine the detector output into a digit string. The hypothesis combination can be formulated as a search problem on a weighted directed graph G, which is a pair (V, E), where V is a set of vertices, and E is a set of edges between the ordered vertices $E = \{(u, v)|u, v \in V\}$. Meanwhile, there is a weight $W_{u,v}$ associated with each edge.

The following procedure can be used to convert the hypothesis lattice into a directed graph.

1. Constructing the node set, V, which consists of all the detected digit boundaries.

For instance, for one detected segment (T_a, T_b), both T_a and T_b will be elements of V.

2. Ranking all the detected boundaries in a time line and adding a edge for each pair of adjacent nodes in the graph in order to guarantee the existence of a path from the start node to the end node.

3. For each detected segment, adding a edge from its start node to its end node.

4. Adding reversal edge to those nodes which are very close to each other (e.g. within 20 ms) or merging these nodes into one node, due to the potential overlap in the detected boundaries.

Given the constructed directed graph, the weight we choose should be consistent with our search criterion. For example, when the search is based on the maximum likelihood criterion, the log-likelihood can be used as the weight. Of course, we can put other score metrics to each edge under a certain criterion.

Finding the best path in a WDG is a well studied problem in computer science and operation research. So finding the best matched string over the detector output lattice is equivalent to finding a path with the maximal weight. The well-known Dijkstra's algorithm can be used to find the best matched path. To further improve the recognition performance by rescoring with other detectors' results, the K-shortest path algorithm can be used to find the K-best digit strings. Figure. 25 is the WDG converted from Figure. 23. Each node in the graph is a detected digit boundary. The number in the node is the time stamp (in 10 ms). Each edge represent a detected digit or a silence edge. The number beside each edge is the frame average log-likelihood. And the red edges are the best path we obtained for the utterance 31o2.

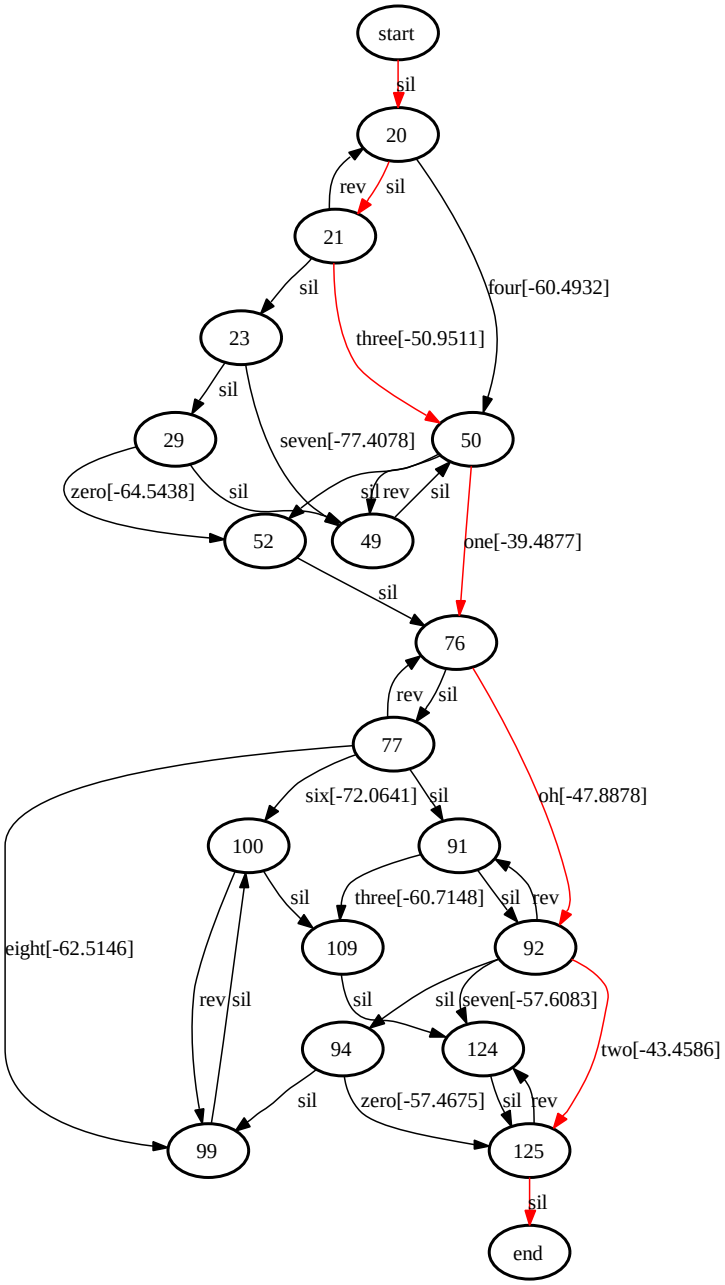

Figure 25: A weighted directed graph.

4.5.3 Digit Recognition Experiments and Result Analysis

All the evaluation experiments were conducted on the TIDIGITS corpus [62]. The digit vocabulary is made of 11 digits, one to nine, plus oh and zero. The training set has 8623 digit strings and the test set has 8700 digit strings. A conventional procedure is used for front-end processing. 12-dimensional MFCC and the log-scaled

energy were extracted for each 10-ms frame. Their first and second order derivatives are also computed for each frame. To conduct cross-corpus evaluation and reduce the channel effects, every element of the feature vector has been normalized with zero-mean and unit variance.

In this experiment, the training set from the TIDIGITS corpus are used to train the whole-word HMM model for each digit. Each HMM has 12 states and use a simple left-to-right topology without state-skip. A state-of-the-art HMM based ASR system and a detection-based ASR system are built for comparison. The conventional HMM based ASR gave a word error rate of 0.48% and the detection-based ASR was slightly worse at 0.73%. So in the matched acoustic condition, the detection-based system can get comparable results as the conventional ASR system.

Now we simulate a real ASR scenario. We purposely introduced a mismatched condition to illustrate the benefits of incorporating knowledge into the detection based ASR system. TIMIT [34] was used for mono-phone model training while the TIDIGITS was down-sampled from 20 KHz to 16 KHz and used for testing. Each mono-phone model is a 3-state left-to-right HMM. A conventional Viterbi-based ASR system and a detection-based ASR system were built for the experiment. The deletion, substitution and insertion errors of step-by-step knowledge-based pruning are shown in Table 1.

The word error rate of the conventional ASR system is 4.54%. For the detection-based ASR system without pruning, it is 6.37%. It's clear that the detection-based system has much more substitution and insertion errors.

Duration Pruning: When we took a look at the recognition results of the detection-based ASR system, we found too many short segments were detected and recognized as words. So the phone-dependent duration constraints can be imposed on the detection results. After pruning with the duration constraints, the word error rate of the detection based ASR system was reduced to 5.03%. The insertion errors were

reduced from 791 to 351, while the deletion errors increase from 167 to 227.

Manner Pruning: We also observed that some confusion pairs are very significant in the word confusion matrix. For example, five/nine (ground-truth/recognized result), five/four, one/nine, eight/three, seven/five, four/oh, etc. Some of these substitution errors can be reduced by manner model based pruning discussed in Section 3.2. The rules used for pruning can be learned from some development data by decision tree. The manner sequence pattern pruning method can generally be used to prune those clear confusions. The overall performance after manner model based pruning is 4.23%. We can see that the substitution errors were reduced from 860 to 620 and the insertion error were reduced from 351 to 302.

Signal Feature Pruning: Signal feature based pruning is often more meaningful and robust. The spectral features of nasal and fricative can be used in five/nine confusion pair. The substitution errors of five/nine were reduced from 51 to 11 by using the low frequency energy ratio and a voicing detector. As for the eight/three confusion pair, the spectrum before the segment /iy/ in three and segment /ey/ in eight are different due to the existence of the fricative /th/ and glides /r/. With a voicing detector and high frequency energy ratio in these two segments, we can reduce the substitution of eight/three from 56 to 24. Similar work can be done on other confusion pairs to reduce those hard confusions. Now the overall performance was improved to 3.74%. The substitution errors have been further reduced (from 620 to 524), while the deletion errors was increased a little (from 258 to 286).

From our experiment results, this kind of signal feature based pruning is very promising. It should be noted that no digit model was used in digit detection and pruning. For reference, if we use the digit-specific models for pruning, the word error rate of the detection-based ASR system is 2.15%. That is better than the result of conventional state-of-the-art ASR system. It shows that even if the acoustic model for detector design is not perfect, we can still have very good recognition performance

by word detection and appropriate pruning strategies. We want to point out that if digit-specific database is used with a new discriminative training algorithm, the string accuracy of TIDIGITS task is 99.33%.

Table 16: Digit recognition results.

	Del.	Sub.	Ins.	Word Err. (%)
Detection W/O Pruning	167	864	791	6.37
W/ Duration Pruning	227	860	351	5.03
W/ Manner Pruning	258	620	302	4.23
W/ Feature Pruning	286	524	260	3.74
Digit-specific Pruning	370	118	126	2.15
Conventional ASR	469	617	211	4.54

4.5.4 KWS Experiment Setup and Result Analysis

Another experiment was conducted on a LVCSR task for arbitrary single word detection. All the evaluation experiments were carried out on the WSJ0 corpus. Both the WSJ0 training set (7132 sentences from 84 speakers) and the TIMIT training set (3696 sentences from 462 speakers) were used in acoustic modeling of context independent monophone models, broad phonetic class models and background models for cross-corpus evaluation. The WSJ0 testing set (Nov92 non-verbalized 5k closed set) consists of 330 sentences from 8 speakers. A conventional procedure is used for front-end processing. To conduct cross-corpus evaluation and reduce the channel effects, every element of the feature vector is normalized with zero-mean and unit-variance [74].

The keywords, including both content and function words, were randomly selected from the original 5k WSJ vocabulary. 30 content and 20 function words had been chosen. The cut-off frequency was set to 8 when selecting keywords to ensure a reliable evaluation.

The performance of a keyword spotting (KWS) system is usually measured using a ROC curve and figure-of-merit (FOM). FOM is an upper-bound estimate of word

spotting accuracy averaged over 1 to 10 false alarms per hour. It's the area under the part of the ROC curve with false alarms from 1 to 10 per hour. In practice, we have little interest in the area beyond 10 false alarms per hour. A keyword is considered successfully detected if the mid-point of the hypothesis fell within the reference time interval. All the hypothesized keywords are sorted with respect to their confidence score, and the probability of detection at each false alarm rate was then computed. An average FOM over all keywords is used as the overall performance measure. Generally speaking, there will be more false alarms with more keywords in the keyword list.

Several comparative experiments have been conducted. The first one was to evaluate the performance of the conventional KWS system under matched (WSJ0 monophone models) and mismatched (TIMIT monophone models) acoustic conditions. It's clear that there is a big performance gap between content words and function words. Even in matched acoustic condition, FOM of content words is two times larger than that of function words, 48.8% versus 22.3% respectively. The performance drop caused by the acoustic mismatch agrees with our expectation. FOM decreased from 48.8% to 42.6% for content words and from 22.3% to 18.4% for function words in matched condition.

Table 17: FOM for conventional method.

	WSJ0 Model	TIMIT Model
Function Words	22.3%	18.4%
Content Words	48.8%	42.6%

The second experiment (as shown in Table 18) was to conduct knowledge-based pruning and rescoring on the output of a conventional KWS system. We can see significant improvements for both content words and function words, under both matched and mismatched conditions. FOM increased from 48.8% to 58.9% for content words in matched condition and similar results are achieved for function words. It manifests the effectiveness of the knowledge-based pruning and rescoring strategy.

Table 18: FOM for conventional method with pruning.

	WSJ0 Model	TIMIT Model
Function Words	29.5%	25.1%
Content Words	58.9%	54.7%

The third experiment (as shown in Table 19) was to conduct knowledge-based pruning and rescoring on the output of the proposed network and filler model selection. For content words, FOM increased from 58.9% in Table 18 to 61.5% in Table 19. This small improvement is attributed to the new network structure. Comparing with the result in Table 17, the performance improvement is significant. For content words, FOM increased from 48.8% to 61.5%.

Table 19: FOM for proposed method with pruning.

	WSJ0 Model	TIMIT Model
Function Words	33.1%	29.7%
Content Words	61.5%	58.3%

It's clear that both the proposed grammar network and the knowledge-based pruning and rescoring strategy are very effective, even with less detailed acoustic model (monophone models) and under mismatched condition (TIMIT models).

4.6 Information Integration by Soft Decision

Another way is to incorporate the attribute information into the LVCSR system by soft decision. A variety of combination schemes exist in the literature of speech recognition, e.g., recognizer output voting error reduction (ROVER) [27] [57], feature concatenation [80], multi-stream model combination [19] [54] and lattice rescoring [97]. These techniques are usually used at different levels of a LVCSR system. Previous research reported performance improvements using one or two of these schemes on different tasks [29] [80] [97]. In this section, we conducted a comparative evaluation of all these typical system combination schemes on a large vocabulary broadcast news transcription task. The effectiveness, advantages and computational cost of

each of these approaches have been investigated and analyzed. We can draw some conclusions from our experiments: (1) Although the MLP feature alone is 5.9% worse than the PLP feature, it does encode complementary information to PLP features; (2) Model combination with independent tree is an effective combination scheme, which consistently outperforms ROVER, feature concatenation, and lattice rescoring.

Four typical combination schemes are briefly described in the following sections. The feature streams investigated in this paper are denoted by $\boldsymbol{o}_{\text{PLP}}^t$ (PLP features), $\boldsymbol{o}_{\text{MFCC}}^t$ (MFCC features) and $\boldsymbol{o}_{\text{MLP}}^t$ (MLP-based phoneme posterior probability features) respectively for time instance t.

4.6.1 Feature Concatenation and Single-stream System

A simple and straightforward method of system combination is to augment the conventional PLP features with the MLP posterior features at the input to a HMM system, i.e., $\boldsymbol{o}^t = [\boldsymbol{o}_{\text{PLP}}^t, \boldsymbol{o}_{\text{MLP}}^t]$, which is a time-synchronous early fusion scheme. Then a single stream HMM-based ASR system is built in the conventional way and all feature streams are modeled jointly. The output probability distribution of state j at time t is usually represented by a Gaussian mixture model (GMM) as shown in Eq. (79):

$$b_j(\boldsymbol{o}^t) = \sum_{m=1}^{M_j} c_{jm} \mathcal{N}(\boldsymbol{o}^t; \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}), \tag{79}$$

where M_j is the number of mixture components of state j, c_{jm} is the weight of the m'th component of state j and $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate Gaussian with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

Feature concatenation usually results in a high-dimensional feature vector $\boldsymbol{o}^t$. Sometimes it is too large to model all feature streams efficiently. When covariance matrices $\boldsymbol{\Sigma}$ are diagonal matrices (and they usually are), each state j has $M_j\,(2\text{dim}(\boldsymbol{o}^t) + 1)$ parameters to be estimated, where $\text{dim}(\boldsymbol{o}^t)$ is the dimension of vector $\boldsymbol{o}^t$.

4.6.2 Model Combination and Multi-stream System

Model level combination is generally formulated as a multi-stream system, which is more flexible and enables separate modeling of multiple information sources, permits different number of states and different number of Gaussian components in each stream. Log-likelihood scores from all feature streams are combined by weighted linear functions. A basic assumption on multi-stream systems is that given the state q^t at any time t, the feature vectors from each of these streams are statistically independent from all other streams. So in the decoding stage, the state output probability distribution is factorized as follows:

$$b_j(\boldsymbol{o}^t) = \prod_{s=1}^{S} \left[\sum_{m=1}^{M_{js}} c_{jms} \mathcal{N}(\boldsymbol{o}_s^t; \boldsymbol{\mu}_{jms}, \boldsymbol{\Sigma}_{jms}) \right]^{w_{js}}, \tag{80}$$

where S is the number of streams in a multi-stream system and the exponent w_{js} is a state-dependent weight for state j of stream s or simply a shared state-independent stream weight w_s. It is used to control the contribution from each of these feature streams and indicate our confidence on each feature stream.

In the case of using diagonal covariance matrices, each state j will have a number of $\sum_{s=1}^{S} M_{js} \left(2\text{dim}(\boldsymbol{o}_s^t) + 1\right)$ parameters to be estimated and $b_j(\boldsymbol{o}^t)$ can be equivalently represented as a single-stream system with $\prod_{s=1}^{S} M_{js} \left(2\text{dim}(\boldsymbol{o}^t) + 1\right)$ parameters. Therefore, a multi-stream system can model the feature streams more accurately than a single-stream system with a similar number of parameters.

Several approaches have been proposed to estimate the parameters of multi-stream systems [19], which consists of two parts: estimation of HMM parameters for each stream and estimation of appropriate stream exponents. For instance, parameters of each stream can be estimated independently such that we have separate models for each feature stream, which means different streams will have independent decision trees and HMM model sets. For the optimal stream weight estimation, there are some techniques discussed in [19] [41] and it is beyond the intention of this paper.

A computationally efficient model combination scheme [19] is employed in this paper for LVCSR tasks, where a primary feature stream (e.g., PLP features) is chosen to build the phonetic decision trees and the initial maximum likelihood models. Then a single-pass retraining is carried out to estimate the HMM model parameters for each stream based on the state occupation probabilities accumulated from the primary feature stream. One advantage of this approach is that only one decision tree and one decoding graph are needed. The stream weights are set to equal for all feature streams in this paper, which had been demonstrated to be as good as state-dependent weights in [19].

4.6.3 Lattice Rescoring

A lattice is a compact representation of the competing hypotheses generated by a decoder. It can be expanded by additional acoustic and language model scores. Usually the lattice oracle word error rate is much smaller than the best path word error rate, which means that it is possible to improve recognition accuracy by re-ranking these competing hypotheses using complementary knowledge sources.

The MLP phoneme (or articulatory attribute) posterior probability was used to rescore word lattices [97]. We followed the exact lattice rescoring scheme proposed in [97]. The phoneme posterior probability is added to the conventional acoustic score by a weighted linear function for each hypothesized word, which corresponds to an arc in the lattice. The rescoring formula used in [97] can be reformulated as Eq. (81):

$$b_j(\boldsymbol{o}^t) \quad = \quad \left[b_j(\boldsymbol{o}^t_{\mathrm{PLP}})\right]^{w_1} \left[f_j(\boldsymbol{o}^t_{\mathrm{MLP}})\right]^{w_2}, \tag{81}$$

where $f_j(\boldsymbol{o}^t_{\mathrm{MLP}})$ simply outputs the phoneme posterior probability corresponding to the hypothesized phoneme at time t based on the aligned state information.

By this reformulation, the difference between lattice rescoring and model combination multi-stream system is clear: in lattice rescoring, the phoneme posterior probability vector MLP^t is directly used in score combination, while in the model

combination system, it is integrated through a probability distribution.

4.6.4 ROVER

Recognizer output voting error reduction (ROVER) [27] is a post-processing scheme for multiple ASR system combination. The rationale behind ROVER is that multiple ASR systems usually exhibit different error patterns, which means that simple majority voting can achieve a lower word error rate than any of the individual systems. The outputs of multiple ASR systems are aligned into a word transition network (WTN) by dynamic programming and then a majority voting is performed for each correspondence set. In contrast to other combination schemes, ROVER works at the output of multiple ASR systems and no acoustic and language models are involved at this stage.

4.6.5 LVCSR Experiment Setup and Result Analysis

A MLP is trained on a set of 3-hour broadcast news speech (2.5-hour for training and 0.5-hour for cross-validation), which is separate from the data sets used in LVCSR experiments. The raw feature used as input of MLP is the TRAP feature, which is the temporal trajectory of log energy from 31 consecutive frames ($\pm$ 15 frames of a central frame) for each critical band. Band-wise mean and variance normalization was also conducted. Then discrete cosine transformation (DCT) is applied to each band to perform dimension reduction and de-correlation. Only the first 10 DCT coefficients and the log-energy of each band are preserved in our experiments. There are 23 critical bands used for 16k Hz speech data. TRAP features were extracted using the trapper software[3].

The topology of the MLP has 253 input units, 800 hidden units and 44 output units that corresponds to 44 phonemes used in LVCSR experiments. The total number of the parameters of the MLP is about 240k. The MLP was trained using the ICSI's

[3]http://speech.fit.vutbr.cz/files/software/trapper.html

QuickNet toolkit[4].

Usually, the output of MLP before the final softmax activation function is used as input to PCA to perform de-correlation. In our experiment, we took the log of the MLP output posterior vector to generate $\boldsymbol{o}^t_{\text{MLP}}$ such that each dimension can be modeled as a Gaussian more accurately. LDA is employed instead of PCA as a discriminant transformation for dimension reduction.

Figure 26 shows the posteriogram of a test utterance, which is a visualization of the MLP features used in system combination experiments. The top panel of the figure is the aligned phoneme sequence and each row corresponds to a sequence of posterior probabilities with respect to a phoneme.

The training and test sets used in LVCSR experiments have 50-hour and 2-hour of English broadcast news speech data respectively. After the MLP was trained, both training and test sets of LVCSR experiments were forwarded through the MLP to get phoneme posterior features. Since the MLP was trained using speaker independent TRAP features, for a fair comparison, no vocal tract length normalization (VTLN) was used in our LVCSR experiments.

Baseline systems were built using PLP and MFCC features. 13 dimensional PLP features were first extracted for each frame and utterance level cepstral mean normalization was performed. Then LDA was used to project 9 contiguous PLP vectors to a 40 dimensional vector, which is the $\boldsymbol{o}^t_{\text{PLP}}$ used in this paper. In addition, 40 dimensional MFCC features $\boldsymbol{o}^t_{\text{MFCC}}$ were extracted for each frame following the same pipeline as PLP features. Phonetic decision trees and maximum likelihood (ML) HMM models were built using $\boldsymbol{o}^t_{\text{PLP}}$ feature stream, which consist of 3k shared quinphone states as the leaves of decision trees and 50k Gaussian components in the ML models. A 4-gram language model and a lexicon with 84K words were used in LVCSR experiments. A dynamic decoder is used in stead of the pre-compiled finite

[4]http://www.icsi.berkeley.edu/Speech/qn.html

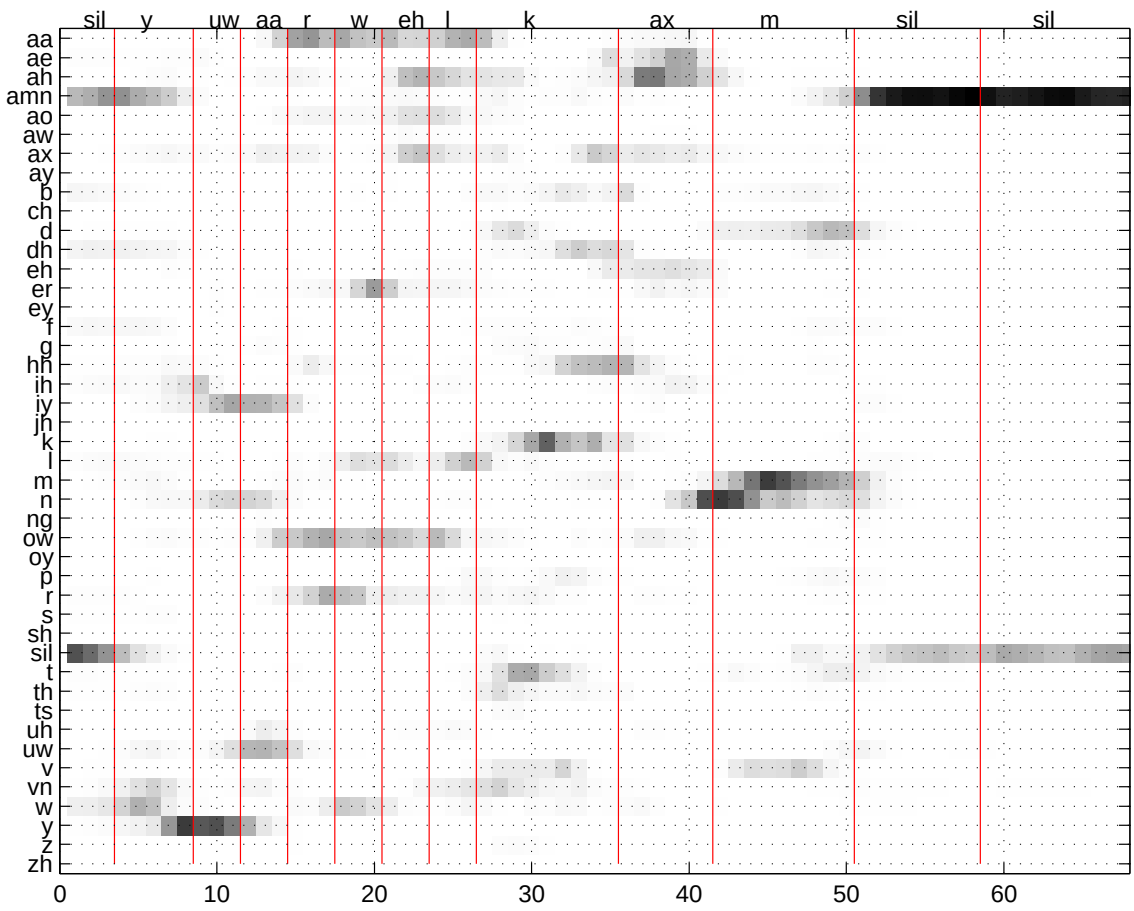

Figure 26: Phoneme posteriogram of sentence "you are welcome".

state decoding graph.

Table 20: WERs of feature concatenation.

	WER (%)
PLP	30.8
MFCC	31.0
TRAPMLP	36.7
PLP-TRAPMLP	32.2

First, we compared the word error rates (WERs) of LVCSR systems using separate and concatenated feature streams. Each feature stream has its independent decision tree and HMM models. In Table 20, TRAPMLP indicates the TRAP MLP feature streams and PLP-TRAPMLP indicates the concatenated feature stream from PLP and TRAPMLP streams. We observed that the best single system is with PLP

features and MFCC can achieve comparable results as PLP features. TRAPMLP system is worse in WER by 5.9%, which is consistent with previous research [29] [80]. The MLP used in this paper was trained with very limited data (2.5-hour), so performance is expected to improve when trained on more data. PLP-TRAPMLP is worse than PLP by 1.4% in WER but better than TRAPMLP by 4.5%.

This experiment is to evaluate the model combination scheme, where each stream has its independent decision tree and HMM models as in section 4.6.5. The WERs for each feature stream is shown in Table 20. Table 21 summarizes the WERs for both model combination and ROVER, where "+" means model combination. We observed that multi-stream decoding is better than ROVER and always get improvements over the PLP baseline system. In addition, the best combination (PLP + MFCC + PLP-TRAPMLP) is 1.9% better in WER than best single system (PLP).

Table 21: WER of independent tree model combination.

	WER (%)
PLP + TRAPMLP	30.7
PLP + MFCC + TRAPMLP	29.8
ROVER	30.1
PLP + MFCC + PLP-TRAPMLP	28.9
ROVER	29.7

In this experiment, the decision tree of TRAPMLP was shared by all feature streams. The HMM models for each stream is obtained by a single pass retraining procedure [19]. From Table 22, we observed that combining three models with shared TRAPMLP tree (32.2%), we get a nice improvement (1.8%) ER over the best single component model (PLP 34.0%). However, with this tree, the PLP model is sub-optimal: 3.2% worse than a decision tree trained using PLP (30.8%) in Table 20.

In this experiment, the decision tree of PLP was shared by all feature streams [19]. It is clear that the MFCC, TRAPMLP and PLP-TRAPMLP systems are worse

Table 22: WER of shared MLP tree.

	WER (%)
TRAPMLP	36.7
PLP	34.0
MFCC	34.2
PLP + MFCC	34.0
TRAPMLP + PLP + MFCC	32.2

than that with their independent trees. In addition, combining three models (PLP + MFCC + TRAPMLP) with a shared PLP tree didn't yield any improvements over the PLP baseline system. However, combining three models (PLP + MFCC + PLP-TRAPMLP) achieved an improvement about 1.3%.

Table 23: WER of shared PLP tree.

	WER (%)
PLP	30.8
MFCC	31.7
TRAPMLP	38.6
PLP-TRAPMLP	32.8
PLP + MFCC + TRAPMLP	31.2
ROVER	30.7
PLP + MFCC + PLP-TRAPMLP	29.5
ROVER	30.2

Lattice rescoring experiments were conducted on speaker adapted systems. We evaluated the combined systems with several configurations, which correspond to some pairs of w_1 and w_2. The experiment results are shown in Table 24. The baseline system has a WER of 24.2% when w_1=0.055 (which is the acoustic score scaling factor used in LVCSR experiments) and w_2=0. When both w_1 and w_2 are 0, only the language model score was used to find the best path and can achieve a WER of 39.2%. When only the MLP posterior score was used to find the best path (w_1=0,w_2=0.05), WER is about 34.8%. When both w_1 and w_2 are set appropriately, we observed a very slight (0.2%) improvement over the PLP baseline system.

Some observations from lattice rescoring experiments are: (1) phoneme recognition

Table 24: WER of lattice rescoring.

w_1	w_2	WER (%)
0.0	0.0	39.2
0.0	0.05	34.8
0.055	0.000	24.2
0.055	0.005	24.0

accuracy could be critical to the effectiveness of lattice rescoring. In our experiments, the MLP based phone recognizer achieved a frame-wise accuracy of 59% on broadcast news; (2) Because the MLP features were obtained from a long-span TRAP features, the phoneme boundaries on the posteriogram as shown in Fig. 26 are not sharp, which caused some errors in lattice rescoring.

This section described a comparative study on system combination schemes for LVCSR tasks. Long-span MLP posterior probability features and conventional short-term cepstral feature are combined using 4 typical combination schemes. Even though the MLP features by themselves are 5.9% worse than PLP, when they are employed in conjunction with conventional cepstral features using model combination with independent tree yields a 1.9% improvement. Simple feature concatenation scheme doesn't work in our experiments and lattice rescoring can achieve a very slight improvement. We observed that multi-stream decoding is better than ROVER.

CHAPTER V

A DETECTION-BASED VIDEO STORY SEGMENTATION SYSTEM

Video broadcast news is hierarchically structured, from frames to shots, scenes, and stories. Figure 27 shows some shots in a video segment, and Figure 28 shows some contiguous frames in a shot. Among these pieces of structure information, story boundaries are of great importance for real-world applications, because the story is a basic unit of many multimedia indexing, retrieval, and management systems.

The task of video story segmentation is to identify the individual news items in a news show, which can be formulated as a sequence segmentation problem. Some studies have been conducted using different knowledge representations and machine learning algorithms based on different design strategies. For instance, the *Informedia* system [39] was one of the early rule-based systems. Some ad hoc rules were designed to combine visual, acoustic, and textual features. Other state-of-the-art video story

Figure 27: Shots in a video segment.

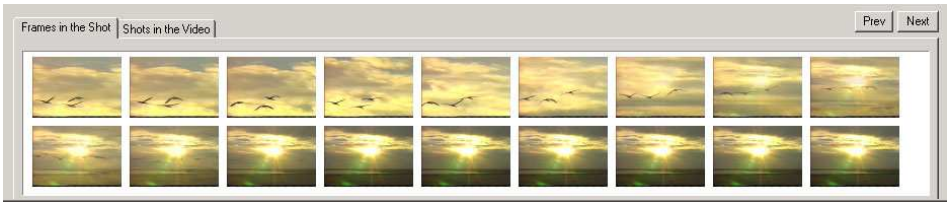

Figure 28: Frames in a video shot.

segmentation systems are based on statistical modeling approaches. Within the Shannon's channel decoding framework for pattern recognition, some algorithms have been investigated. For instance, the broadcast news was modeled with a hidden Markov model (HMM) [11] [14] or a probabilistic context-free grammar (PCFG) [45] such that the story boundaries and other complicated structure information can be obtained by a decoding procedure or a parsing tree. For a HMM or PCFG based system, tokenization of the multimedia stream is crucial. The story segmentation performance greatly depends on the selection of the tokens used to represent the video structures. For example, 17 predefined shot categories are used to capture the structure information [11]. Some of them are program specific logos, such as "SPORT" and "TOP". These tokens are helpful in finding the structures of broadcast news videos. However, its limitation is obvious: the production rules and the style of the broadcast news video vary over programs and time. So a complete and accurate channel specification that is the key to the success of Shannon's channel decoding paradigm cannot be easily realized for such a complex system, which deals with so many diverse sources of information that cannot be handled in an integrated manner.

Another problem of the existing story segmentation systems is that the optimization criteria in the training phase is inconsistent with the performance metric used in the evaluation phase. This problem becomes more severe when dealing with imbalanced data, where negative instances will generally dominate both the training and evaluation data sets. For example, in video story segmentation, the portion of the true story boundaries in all the candidate story boundaries is about 4%. The widely used up-sampling with replacement of positive instances or down-sampling of negative samples will bring some difficulties. When re-sampling is finished, the two group of instances tend to have equal number of examples or have a pre-fixed ratio. However, the class prior information was lost.

In this chapter, we present a detection-based video story segmentation system,

which consists of multi-modal event detection and a discriminative evidence fusion scheme. The maximum figure-of-merit (MFoM) learning approach stated in chapter 3 is used for discriminative evidence combination, which directly optimizes the performance metrics used in segmentation performance evaluation, such as precision, recall, and F1 measure. Another advantage from using rMFoM learning is that in previous studies, this approach demonstrated good performance for imbalanced data as well [31]. Our experimental result on TRECVID 2003 dataset also showed its effectiveness for video story segmentation.

5.1 Overview of Detection-based Segmentation System

Inspired by the studies of human vision [109] and auditory perception [3], a detection-based framework is proposed to deal with heterogeneous multi-modalities of broadcast news video. By decomposing a difficult problem into small pieces, a divide-and-conquer strategy would provide attractive and feasible solutions. Each subproblem could be solved with different design instead of a single feature or a single likelihood computation. By solving smaller pieces one by one, the evidence collected from subproblems will be combined to obtain a solution for the complex system.

For video story segmentation with many diverse and heterogeneous information sources, a detection-based framework provides a natural and intuitive solution, where story segmentation is divided into two steps: event detection and evidence fusion. First, a collection of possible story boundaries were detected as candidates. And then the evidence around each candidate point was collected and some fusion approaches were used to combine the heterogeneous information. For instance, support vector machine (SVM) [43] and maximum entropy (MaxEnt) model [42] have been successfully investigated. Figure 29 shows a detection-based design for video story segmentation. From the context of each candidate boundary, there is rich of information from heterogeneous knowledge sources, such as anchor information, key-phrases,

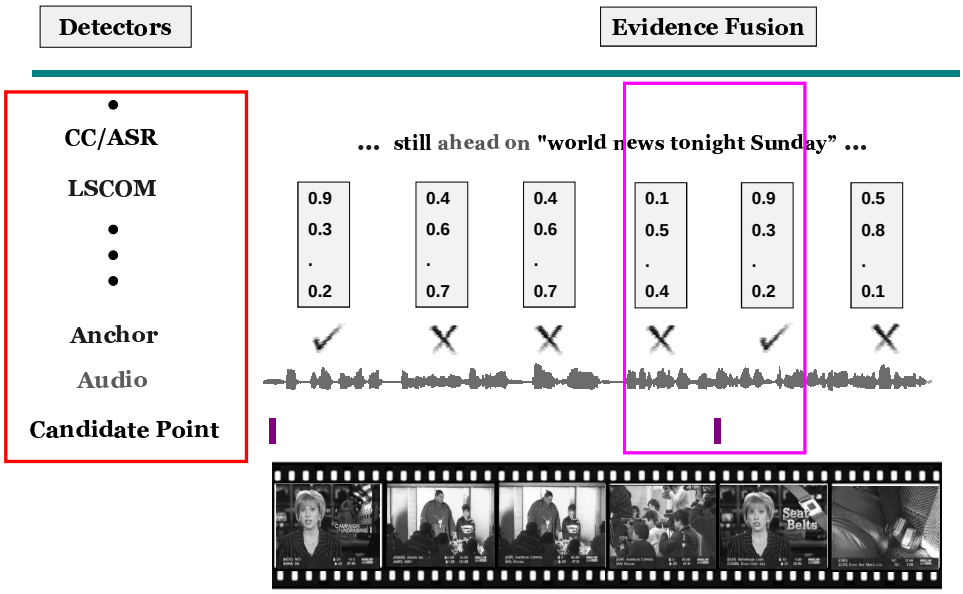

Figure 29: Detection-based video story segmentation system.

and audio events.

5.2 Knowledge Sources and Event Detection

In the knowledge hierarchy of broadcast news video, a shot is the basic unit of a story segmentation system. Our experimental results show that 93.3% of the true story boundaries were covered by shot boundaries. Shot boundary detection can be regarded as a "solved" problem [65]. We take the union of shot boundaries and long audio pauses as candidate points but remove duplications within 2.5-second fuzzy window. Our study showed these two sets of candidate points account for 98% of the true story boundaries in video broadcast news.

At each candidate point of story boundaries, visual, acoustic, and textual features could be extracted and different multi-modal event detectors will be constructed. It is natural to design detectors for different modalities separately. The criteria and techniques discussed in Chapter 2 will be employed to conduct primitive feature detection.

Detectors are basic units of a detection-based automatic video analysis system.

They can be implemented at different levels such as face detectors, anchor detectors, speech/music detectors and so on.

In addition, automatic indexing of video data requires knowledge at multiple levels. Therefore, a detection-based framework is a natural way of knowledge representation and meta information extraction. In broadcast news video story segmentation, detection and verification are performed for each knowledge source, such as anchor shot, audio type information, image annotation, key-phrase information and so on.

5.2.1 Unsupervised Anchor Shot Detection

Anchor shot is a crucial primitive feature of broadcast news videos. We are aiming to find all the shots with one or two anchor persons in a studio background. Our experiments on a 17-hour broadcast news video dataset show that if an anchor shot location is treated as a news story boundary, this feature alone can achieve an accuracy of 61.7% in story segmentation. Similar results have been verified by many other studies. Research work on significant feature selection for story segmentation using an information gain criterion also shows that anchor shot is the most important feature for video story segmentation [14]. A probabilistic confidence score can also be obtained for each shot instead of a hard decision.

Previous studies on anchor shot detection include both supervised and unsupervised methods [92] [106]. Almost all systems use only visual features and face information from keyframes. To utilize the spatial information in anchor shots, some researchers employed pattern matching with several predefined spatial structures [107]. A supervised system using SVM gave an average accuracy of 91.3% on the TRECVID 2005 dataset [106], and the unsupervised anchor shot detection usually results in an average accuracy of about 60%-80% [14]. Although supervised systems have better performance for a specified channel and program, and can be used in on-line processing, its limitation is obvious: the production rules and the style vary dramatically

over channels and time.

Generally, unsupervised anchor shot detection systems perform clustering on the shots with detected faces using visual features, such as a color histogram and texture information [14]. In another unsupervised system, graph-theoretical clustering (GTC) with minimum spanning tree (MST) is used with only visual features and a face detector [92]. In all these unsupervised systems, cues from the audio track were ignored. Part of the reason is that it is not trivial to find suitable representations for multi-modalities to integrate these heterogeneous features into a unified framework. In this section, a novel unsupervised learning approach to represent multi-modal features in a unified and systematic manner is proposed. Spectral clustering with multi-modal features from video, audio, and high level information is investigated thoroughly.

Spectral clustering originated from graph partitioning based on spectral graph theory and has been studied intensively and extensively in machine learning communities [15] [82]. The core idea can be formulated as follows.

Given n vectors, $X = (x_1, x_2, \cdots, x_n), x_i \in \mathbb{R}^d$, a weighted undirected graph $G = (V, E)$ is constructed to encode the neighborhood structure of X, where V is the vertex set and E is the edge set. Each edge $e_{i,j}$ connecting nodes i and j is associated with a weight $d(i, j) > 0$. An affinity matrix A, is formed for G to represent the pairwise similarity. In practice, the affinity matrix is often obtained using kernel tricks to project the data into a high dimensional feature space as follows and a Gaussian kernel is the commonly used one,

$$A_{ij} = \exp(-\frac{d(i,j)}{\sigma^2}), \tag{82}$$

where σ is the size of the Gaussian kernel and used as a scaling factor.

There are many variants of spectral clustering algorithms. The major difference is in the construction of the affinity matrix. In our study, we followed Ng's algorithm [82] for single affinity matrix and extend it to work with multiple affinity matrices.

1. Define $D = \text{diag}(d_1, d_2, \cdots, d_n)$ to be the degree matrix of A, here $d_i = \sum_{j=1}^{n} A_{ij}$, and construct a normalized affinity matrix L as follows:

$$L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{83}$$

2. Find the k largest eigenvectors of L, and form a matrix U by stacking the eigenvectors in columns, $U = [u_1, u_2, \cdots, u_k] \in \mathbb{R}^{n*k}$.

3. Form a matrix R from U by normalizing each row of U to have unit length. A row vector in R is a new feature vector associated with each node. Now all the nodes are on a unit sphere in the spectral space spanned by the k largest eigenvectors.

4. Cluster the rows of R into k clusters with a k-means algorithm or any other clustering algorithms.

Spectral clustering has many advantages over conventional clustering algorithms. First, kernel techniques are used to project the data into a high-dimensional feature space in which the clusters can be more spatially distinct and compact. Second, spectral clustering is also a nonlinear dimensionality reduction method. The actual clustering process is performed in a low-dimensional spectral space spanned by the first k largest eigenvectors of the normalized affinity matrix. It's more efficient and robust for initialization. Third, theoretical analysis shows that spectral decomposition can reveal the block structure of the affinity matrix, which is related to the number of intrinsic clusters [82]. Finally, in spectral clustering, only a distance matrix is needed instead of individual samples for each attribute or a centroid of a cluster. For instance, it's hard to find a centroid of a group of audio segments with different durations. So it's difficult to integrate acoustic features into conventional k-means based clustering algorithms, whereas the shot-wise distance matrix from acoustic features or other attributes can be constructed easily. These properties make spectral clustering an

ideal choice for integrating multiple heterogeneous information encoded in attribute distance matrices for unsupervised anchor shot detection.

A subset from TRECVID 2004 [56] is used for the anchor shot detection evaluation. It consists of 34 video clips with a total length of about 17 hours. The data are CNN and ABC broadcast news videos from 1998. For a reliable evaluation, all the anchor shots were manually labeled for this dataset. The performance of an anchor shot detection system is usually evaluated with precision and recall measures used in information retrieval. Meanwhile, a single F1 score that combines the recall and precision is also used for performance comparison.

Table 25 shows the performance of the unsupervised and supervised anchor shot detection system. It's clear that the performance of our unsupervised system (F1 = 0.871) approaches the performance of the supervised system (F1 = 0.902). When visual features are combined with acoustic features with an appropriate weight, the unsupervised detection performance can be improved dramatically. It demonstrates the effectiveness of combining heterogeneous features. Experiment results show that such an integration can greatly improve the performance of the unsupervised system, and the performance of the unsupervised system can approach the performance of the supervised system.

Table 25: Anchor shot detection results.

	Precision	Recall	F1
unsupervised	86.5%	87.6%	0.871
supervised	95.1%	85.8%	0.902

5.2.2 LSI-based AIA Detectors

Semantic concept detectors can help to bridge the semantic gap between the low level features and the high level semantics. Previous studies showed that the semantic concepts provide important information of story boundaries [11]. In addition, the dynamic patterns of semantic concepts showed the structure of a broadcast news

video. Large-scale concept ontology for multimedia (LSCOM) [53] is designed to provide a dictionary of semantic units for general purpose indexing and retrieval of video. And these concepts were used for broadcast news story segmentation in our study.

Forty-three semantic concepts were manually selected and these concepts are shown to be closely related to the story boundaries. For each candidate point, a change of the detected concepts often indicates a potential story boundary.

A latent semantic indexing (LSI)-based automatic image annotation (AIA) system was constructed as described in [9]. This system was verified to perform very well in the AIA task for the Corel photo dataset. As the training and validation set, we used the TRECVID 2005 development set for which all the LSCOM semantic concepts were annotated [53]. Every shot in the TRECVID 2003 used for the story boundary segmentation task was then annotated with the AIA-based detectors. One thing to note is that the video data in TRECVID 2003 and in TRECVID 2005 have large mismatches in video qualities. So instead of creating a hard-decision for each shot, a confidence measure is given by each concept detector. The confidence vectors are combined with the proposed discriminative fusion method.

5.2.3 Audio Type Detection

There are various types of audio signals in the audio track of broadcast news video, e.g, speech, music, silence, and speech with music background, etc. Both the type of audio signal and the change of the audio type imply important information for story segmentation. In our study, each audio type is modeled with a hidden Markov model (HMM) and each shot is represented by a confidence vector of audio type. This is similar to the confidence vectors in AIA detectors.

5.2.4 Cue-phrase Detection

Intuitively, there are some cue phrases around the story transition. The automatic speech recognition (ASR) transcriptions were used to detect the cue phrases within each shot period. To extract a list of cue phrases, we compute the N-gram lexical sequences around reference story boundaries. From this list, 16 transition word sequences are manually selected, e.g., "ABC news", "CNN", "thanks for watching", "coming up", and "ahead on", etc. To be suited in the evidence fusion framework, the detected cue phrases are converted into a score according to the occurrence frequencies. With more cue phrases detected within a shot, the confidence is higher.

5.2.5 Speaker Change and Long Pause Detection

Story transitions in broadcast news video are usually accompanied with significant pause or silence. In addition, speech prosody can contribute to the detection of story breaks, with speaker pause duration often being the most important feature. We usually assume a larger pause between stories than between the sentences of a story. Therefore, we use an energy-based voice-activity detector (VAD) to extract the duration of all the silences in the audio channel. And then we identified the longest silence fragment immediately preceding a sentence, and used this duration as a feature.

5.3 Discriminative Evidence Integration

When the evidence from diverse knowledge sources around each candidate story boundary is available, some fusion strategies could be employed to obtain a final decision by integrating heterogeneous sources. A discriminative fusion scheme, maximum figure-of-merit (MFoM) approach, is used for feature and classifier combination. Unlike other fusion approaches such as maximum entropy (MaxEnt) method that maximizes the likelihood, and SVM that maximize the margin between decision

boundaries, the maximum figure-of-merit (MFoM) learning algorithm directly optimizes the performance metrics used in video story segmentation evaluation: precision, recall, and F1 measure [31].

5.3.1 Maximum Figure-of-merit (MFoM) Learning

The maximum figure-of-merit learning approach [31] approximates the four terms in the contingency table with a differentiable function with regards to the model parameters and directly optimizes the performance metrics. It allows quite a bit of flexibility in choosing the discriminant functions for each class. The discriminant functions $f(\mathbf{x};\mathbf{w})$ can be linear discriminant functions (LDF), quadratic discriminant functions (QDF), or complicated probabilistic discriminant functions such as Gaussian mixture model (GMM) and HMM. With the help of the discriminant functions for both positive (boundary) and negative classes (non-boundary), a misclassification measure $d(\mathbf{x};\mathbf{w})$ can be used to measure the correctness of a decision at a single candidate point. Here $\mathbf{x}$ and $\mathbf{w}$ are evidence vector and parameters of the discriminant functions respectively. $\mathrm{X}_{\mathtt{pos}}$ and $\mathrm{X}_{\mathtt{neg}}$ are training instances from positive and negative classes.

$$d(\mathbf{x};\mathbf{w}) \;=\; \begin{cases} f_{\mathtt{neg}}(\mathbf{x};\mathbf{w}_{\mathtt{neg}}) - f_{\mathtt{pos}}(\mathbf{x};\mathbf{w}_{\mathtt{pos}}), & \mathbf{x} \in \mathrm{X}_{\mathtt{pos}} \\ f_{\mathtt{pos}}(\mathbf{x};\mathbf{w}_{\mathtt{pos}}) - f_{\mathtt{neg}}(\mathbf{x};\mathbf{w}_{\mathtt{neg}}), & \mathbf{x} \in \mathrm{X}_{\mathtt{neg}} \end{cases} \tag{84}$$

The role of a loss function $l(\mathbf{x};\mathbf{w})$ is to use a smooth function to approximate the errors obtained on a dataset. And the sigmoid function is the most widely used one. Here α control the steepness of the curve and β control the covered area of the loss function.

$$\ell(\mathbf{x};\mathbf{w}) \;=\; \frac{1}{1+\exp\left(\beta - \alpha * d(\mathbf{x};\mathbf{w})\right)} \tag{85}$$

The performance metrics such as error rate, precision, recall, and F1 measure can be calculated from the contingency table as shown in Table 26. Here, true positive

(TP), false positive (FP), false negative (FN), and true negative (TN) are approximated by the loss function for each training sample.

Table 26: Contingency table.

	test (+)	test (-)
+	$\text{TP} \approx \sum_{\mathbf{x} \in \text{X}_{\text{pos}}} (1 - \ell(\mathbf{x}; \mathbf{w}))$	$\text{FN} \approx \sum_{\mathbf{x} \in \text{X}_{\text{pos}}} \ell(\mathbf{x}; \mathbf{w})$
-	$\text{FP} \approx \sum_{\mathbf{x} \in \text{X}_{\text{neg}}} \ell(\mathbf{x}; \mathbf{w})$	$\text{TN} \approx \sum_{\mathbf{x} \in \text{X}_{\text{neg}}} (1 - \ell(\mathbf{x}; \mathbf{w}))$

Given the approximation of the four terms in a contingency table, the F1 measure can be approximated as follows:

$$\text{F1} \quad = \quad \frac{2\text{TP}}{\text{FP} + \text{FN} + 2\text{TP}}. \tag{86}$$

5.3.2 Parameter Initialization and Update

One issue that needs to be considered is the initialization of the optimization procedure. Because the approximated F1 measure is a non-convex function with regards to the model parameters, there is no guarantee of finding a global optimal solution. And the obtained local optimal solution greatly depends on the initial value of the parameters. The initialization in our experiments is performed using the expectation maximization (EM) algorithm [22] for GMM and perceptron algorithm [23] for LDF. The parameter update can be conducted using batch gradient descent method, or generalized probabilistic descent method, or quasi-Newton optimization techniques (e.g., L-BFGS).

For each candidate point, there are four evidence scores from different detectors. For example, if the shot right after the candidate point is detected as an anchor shot with a high confidence and there are some cue phrases detected right before the candidate point, this candidate point is very likely to be a true story boundary. When evidence is detected with a hard-decision, some logical rules can be deduced and constructed for video story segmentation. In this paper, GMM discriminant

functions are used in AIA feature fusion and LDFs are used in story segmentation fusion. The MFoM learning approach is used to improve the performance of the final decision. This data-driven approach will assign a weight for each evidence and an offset from both the positive and negative instances. It allows quite a bit of flexibility in evidence detector design. When new detectors are built, their relative importance for final decision making will be learned automatically and discriminatively from data. The inference step becomes really straightforward.

5.4 Experiment Setup and Result Analysis

All experiments were conducted on a standard benchmark dataset. The complete dataset of TRECVID 2003 [98] is used for story segmentation evaluation. It consists of about 110 video clips for development and another 105 video programs for evaluation. Each video clip has a length of about 30 minutes. The data are CNN and ABC broadcast news video of year 1998. We have conducted our experiments using the methodology proposed by TRECVID [98].

The shot segmentation, keyframe extraction, and ground-truth story segmentation were provided by LDC [86]. The audio track was demultiplexed from the MPEG stream with 16 KHz sampling rate and 16 bits. Mel frequency cepstral coefficient (MFCC) features were extracted from audio signals in the audio type detector design [78].

The performance of a story segmentation system is usually evaluated with precision and recall. Meanwhile, a single F1 measure, which is the harmonic mean of the recall and precision, is also used for performance comparison. According to the guideline of TRECVID 2003, when conducting performance evaluation, each reference boundary was expanded with a fuzziness factor of 5 seconds in each direction, resulting an evaluation interval of 10 seconds.

5.4.1 Comparison with SVM Fusion

The first experiment is to compare the performance of the MFoM fusion method with the SVM fusion method using the AIA features. For each candidate point, a confidence vector of 86 dimension is constructed using the AIA detector output for the shot right before and after the candidate boundary. Two fusion strategies have been investigated. The first one is a SVM fusion approach using the LIBSVM [12] toolkit. And the second one is the MFoM scheme. In the MFoM scheme, both the positive class (boundary) and negative class (non-boundary) are modeled by a GMM with 2 mixtures. Figure 30 shows the result of story segmentation using this AIA confidence vector. It's clear that MFoM significantly outperformed the SVM fusion method. The F1 measure from the MFoM scheme is about 0.51 and the F1 measure from SVM is about 0.44.

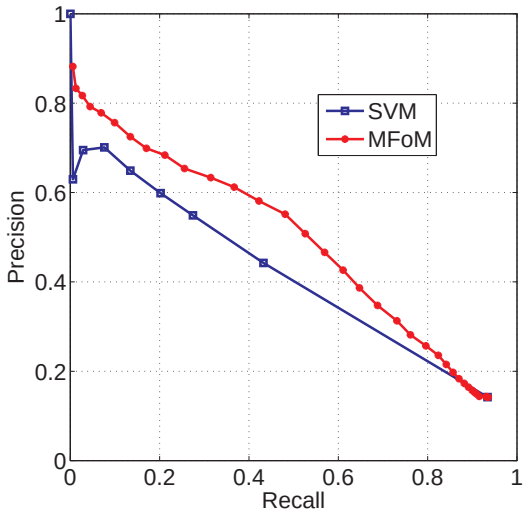

Figure 30: Comparison of SVM fusion and MFoM fusion.

5.4.2 Heterogeneous Information Source Fusion

The second experiment is to demonstrate the effectiveness of MFoM fusion scheme with more heterogeneous information sources. Table 27 shows the performance of story segmentation under different combinations. The upper part of the table shows

the performance using individual detectors. For instance, using only AIA detectors achieves a F1 measure of 0.514, while using only anchor detector can achieve a F1 measure about 0.605. It's also clear that anchor detector has a high precision for story segmentation. It means that the evidence from the anchor detector is a reliable cue. Nevertheless, audio type detectors demonstrated a high recall. It means that audio type changes can cover most of the true story boundaries. These kinds of complementary information gives the detection-based story segmentation system plenty of room for performance improvement. In this detection-based framework, when more and more evidences are available, the system performance can be improved in an additive manner in terms of F1 measure. For example, when the cue phrase detector was combined with the anchor detector, even a simple "OR" operation can improve the F1 measure by 1% from a high performance baseline system. Similar experiments have been done with AIA detectors and audio type detectors in the bottom part of Table 27. It's obvious that as more evidences were combined using MFoM scheme, the recall was greatly increased while the precision was decreased a little.

Table 27: Performance of story segmentation.

	Precision	Recall	F1
Text (T)	0.382	0.208	0.269
Long Pause (P)	0.633	0.296	0.403
Audio (A)	0.194	0.771	0.310
AIA (V)	0.552	0.481	0.514
anchor	0.780	0.494	0.605
anchor + T	0.762	0.520	0.618
anchor + T + V	0.753	0.552	0.637
anchor + T + V + A	0.739	0.581	0.651
anchor + T + V + A + P	0.718	0.613	0.661

5.5 Summary

In this chapter, we presented a detection-based framework for pattern recognition and demonstrated its application to video broadcast news story segmentation. In the

proposed approach, event detectors and fusion schemes can be designed and optimized separately. Therefore, it offers quite a bit of flexibility in system construction and performance improvement. With more and more event detectors available, system performance can be improved in an additive manner. This study also presents a set of multi-modal event detectors built with different signal processing and machine learning methods, and a discriminative fusion method for video story segmentation, which directly optimizes F1 measure.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

The objective of this dissertation is to present a detection-based pattern recognition framework and demonstrate its applications in automatic speech recognition and broadcast news video story segmentation. This research was motivated by the studies of human vision and auditory perception. In contrast to the expectation-driven top-down frameworks for pattern recognition, a detection-based framework first detects a collection of primitive features; then, it constructs the domain-specific knowledge hierarchy level-by-level by information integration schemes. Task-specific knowledge and context information were incorporated into this framework as additional features at any stage. Domain-specific information was also used to detect low-level features using knowledge-guided signal processing methods. In some cases, evidence fusion could be conducted in an incremental manner, which means that when more evidence and constraints are available, the hypotheses will become sharper and more focused until a consistent decision can be made.

6.1 Contributions of This Dissertation

This dissertation presents the basic principles, criteria, and techniques for detector design and hypothesis verification based on the statistical detection and decision theory. Two novel detection algorithms are proposed in this proposal: a semi-supervised learning and a discriminative detection algorithm. The knowledge hierarchy of a detection based framework is task-dependent and domain-dependent knowledge is required to design low-level attribute and event detectors.

For evidence fusion, several fusion strategies have been investigated: feature concatenation, model combination, maximum entropy fusion, and a novel proposed regularized maximum-figure-of-merit (rMFoM) approach as discriminative evidence fusion. A detection-based speech recognition recognition system and a detection-based video story segmentation system demonstrate effectiveness and promising results from our experimental studies.

Some of the expected contributions from this dissertation are highlighted in the following and most of my research work have been published in [70] [74] [71] [73] [72] [68].

1. **A detection-based pattern recognition framework** It was motivated by the studies of human vision and auditory perception model and we are aiming to mimic the human being's pattern recognition process. It decomposes a complex system into smaller pieces, solves each subproblem one by one effectively and reliably, and then forms the solution to the original problem. The detection-based framework is proposed to deal with the uncertainties and complexities of real-world applications.

2. **Investigation of real-world applications** A detection-based ASR system and a detection-based broadcast news video story segmentation system have been implemented to demonstrate the basic principles and general techniques for detector design and evidence fusion, which is different from conventional LVCSR systems. Our study shows that the detection-based framework itself can achieve promising results. In addition, the low-level attributes could provide complementary information for the state-of-the-art LVCSR systems.

3. **A novel supervised robust detector design approach** A point on the precision-recall or ROC curve corresponds to an operating point of a detector. By maximizing the area under the P-R or ROC curve, the proposed robust detector

design approach can achieve the best overall performance over all practical operating points.

4. **A novel semi-supervised model-based detector design approach** A semi-supervised algorithm is proposed to leverage both the limited amount of labeled data and huge amount of unlabeled data. In some experiments, this approach illustrated very promising performance.

5. **A novel regularized maximum-figure-of-merit (rMFoM) fusion strategy** This study is an extension of the MFoM learning approach. By adding a regularization term, this approach demonstrated very good generalization capability. In addition, this approach has been extended to general semi-supervised learning conditions, theoretical analysis and experimental results demonstrate the effectiveness of this approach.

6.2 Future Research Problems

The research work in this dissertation is not meant to claim a complete solution to all related problems. I presented a general framework and some common techniques that could be used in other real-world pattern recognition applications. The studies presented in this dissertation are expected to contribute to many research areas of pattern recognition, such as automatic speech recognition, video analysis, and computer vision. For future research, I believe we should look into the following aspects:

1. **Detector Design** As stated in this dissertation, detector design for primitive feature detection is task-dependent. Accurate and comprehensive domain-specific knowledge will help both detector design and evidence fusion. The basic principles, criteria , and general techniques have been discussed in this dissertation. More studies could be done in both general detection techniques and task-specific detection approaches.

2. **Integration Schemes** We have presented some common strategies for information integration. However, no scheme can always achieve the best performance for all situations. More fusion strategies could be explored both theoretically and experimentally, especially for real-world applications.

3. **Real-world Applications** In this dissertation, two real-world pattern recognition applications have been investigated. In the future, I expect the detection-based pattern recognition framework could be used in other computer vision and audio processing applications to further validate its effectiveness.

REFERENCES

[1] Abney, S., "Understanding the Yarowsky algorithm," *Comput. Linguist.*, vol. 30, no. 3, pp. 365–395, 2004.

[2] Agarwal, S., Awan, A., and Roth, D., "Learning to detect objects in images via a sparse, part-based representation," *IEEE Trans. on PAMI*, vol. 26, pp. 1475–1490, 2004.

[3] Allen, J. B., "How do humans process and recognize speech?," *IEEE Trans. SAP*, vol. 2, pp. 567–577, 1994.

[4] Berger, A. L., Pietra, S. D., and Pietra, V. J. D., "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[5] Bishop, C. M., *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[6] Bishop, C. M., *Pattern Recognition and Machine Learning*. Springer, August 2006.

[7] Breiman, L., "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.

[8] Burges, C. J., "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[9] Byun, B., Ma, C., and Lee, C.-H., "An experimental study on discriminative concept classifier combination for TRECVID high-level feature extraction," in *Proc. of ICIP*, 2008.

[10] Casella, G. and Berger, R. L., *Statistical Inference*. Duxbury Press, 2001.

[11] Chaisorn, L., Chua, T.-S., Lee, C.-H., and Tian, Q., "A hierarchical approach to story segmentation of large broadcast new video corpus," in *Proc. of ICME*, 2004.

[12] Chang, C.-C. and Lin, C.-J., *LIBSVM: a library for support vector machines*, 2001.

[13] Chapelle, O., Schölkopf, B., and Zien, A., *Semi-supervised Learning*. MIT Press, 2006.

[14] Chua, T.-S., Chang, S.-F., Chaisorn, L., and Hsu, W., "Story boundary detection in large broadcast news video archives: techniques, experience and trends," in *Proc. of ACMMM*, pp. 656–659, 2004.

[15] Chung, F. R. K., *Spectral Graph Theory*. Amer. Math. Society, 1997.

[16] Conn, A. R., Gould, N. I. M., and Toint, P. L., *Trust-Region Methods*. SIAM, 2000.

[17] Cover, T. and Thomas, J., *Elements of Information Theory*. Wiley and Sons, New York, 1991.

[18] Crandall, D., Felzenszwalb, P., and Huttenlocher, D., "Spatial priors for part-based recognition using statistical models," in *Proc. of CVPR*, 2005.

[19] Cui, X., Xue, J., Xiang, B., and Zhou, B., "A study of bootstrapping with multiple acoustic features for improved automatic speech recognition," in *Proc. InterSpeech*, 2009.

[20] Davis, J. and Goadrich, M., "The relationship between precision-recall and ROC curves," in *In* Proc. of ICML, 2006.

[21] Dempster, A. P., "A generalization of bayesian inference," *Journal of the Royal Statistical Society*, vol. 30, pp. 205–247, 1968.

[22] Dempster, A. P., Laird, N. M., and Rubin, D. B., "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.

[23] Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern Classification*. Wiley-Interscience, 2000.

[24] Elkan, C., "The foundations of cost-sensitive learning," in *Proc. of IJCAI*, 2001.

[25] Eysenck, M. W. and Keane, M. T., *Cognitive Psychology: A Student's Handbook*. Psychology Press (UK), 2000.

[26] Fawcett, T., "ROC graphs: Notes and practical considerations for data mining researchers," tech. rep., HP Labs, 2003.

[27] Fiscus, J., "A post-processing system to yield reduced word error rates: recogniser output voting error reduction (ROVER)," in *Proc. of ASRU*, pp. 347–354, 1997.

[28] Forsyth, D. A. and Ponce, J., *Computer Vision - A Modern Approach*. Prentice Hall, 2003.

[29] Fousek, P., Lamel, L., and Gauvain, J., "Transcribing broadcast data using MLP features," in *Proc. InterSpeech*, 2008.

[30] Gao, S., Lee, C.-H., and Lim, J.-H., "An ensemble classifier learning approach to ROC optimization," in *Proc. of ICPR*, 2006.

[31] Gao, S., Wu, W., Lee, C.-H., and Chua, T.-S., "A MFoM learning approach to robust multiclass multi-label text categorization," in *Proc. of ICML*, 2004.

[32] Gao, S., Wu, W., Lee, C.-H., and Chua, T.-S., "A maximal figure-of-merit (MFoM)-learning approach to robust classifier design for text categorization," *ACM Trans. Inf. Syst.*, vol. 24, no. 2, pp. 190–218, 2006.

[33] Gao, S., Wu, W., Lee, C.-H., and Chua, T.-S., "A maximal figure-of-merit learning approach to text categorization," in *Proceedings of ACM SIGIR*, pp. 174–181, 2003.

[34] Garofalo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S., and Dahlgren, N. L., "DARPA TIMIT acoustic phonetic continuous speech corpus CD-ROM," tech. rep., NIST, 1993.

[35] Gauvain, J.-L. and Lee, C.-H., "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. SAP*, vol. 2, pp. 291–298, 1994.

[36] Gazzaniga, M., Ivry, R., and Mangun, G., *Cognitive Neuroscience: The Biology of the Mind*. W.W. Norton, 1998.

[37] Gunawardana, A., Mahajan, M., Acero, A., and Platt, J. C., "Hidden conditional random fields for phone classification," in *in Interspeech*, pp. 1117–1120, 2005.

[38] Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer New York, 2 ed., 2009.

[39] Hauptmann, A. and Witbrock, M., "Story segmentation and detection of commercials in broadcast news video," in *Proc. of Advances in Digital Libraries*, 1998.

[40] Heisele, B., Serre, T., Pontil, M., and Poggio, T., "Component-based face detection," in *Proc. of CVPR*, 2001.

[41] Hernando, J., "Maximum likelihood weighting of dynamic speech features for CDHMM speech recognition," in *Proc. ICASSP*, pp. 1267–1270, 1997.

[42] Hsu, W. and Chang, S.-F., "A statistical framework for fusing mid-level perceptual features in news story segmentation," in *Proc. of ICME*, 2003.

[43] Hsu, W., Kennedy, L. S., Chang, S.-F., Franz, M., and Smith, J. R., "Columbia-IBM news video story segmentation in TRECVID 2004," tech. rep., Columbia University, 2005.

[44] Huang, X., Acero, A., and Hon, H.-W., *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall, 2001.

[45] Jacobs, A., Ioannidis, G. T., Christodoulakis, S., Moumoutzis, N., Georgoulakis, S., and Papachristoudis, Y., "Automatic, context-of-capture-based categorization, structure detection and segmentation of news telecasts," in *DELOS Conference*, 2007.

[46] Joachims, T., "Transductive inference for text classification using support vector machines," in *in Proc. of ICML*, pp. 200–209, 1999.

[47] Juang, B.-H., Chou, W., and Lee, C.-H., "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 5, pp. 257–265, 1997.

[48] Juang, B.-H. and Katagiri, S., "Discriminative learning for minimum error classification," *IEEE trans. on Signal Processing*, vol. 40, pp. 3043–3054, Dec 1992.

[49] Juneja, A. and Espy-Wilson, C., "Segmentation of continuous speech using acoustic-phonetic parameters and statistical learning," in *Proc. of ICONIP*, 2002.

[50] Katagiri, S., Juang, B.-H., and Lee, C.-H., "pattern recognition using a family of design algorithm based upon the generalized probabilistic descent method," *IEEE Trans. SAP*, vol. 86, pp. 2345–2373, 1998.

[51] Kawahara, T., Lee, C.-H., and Juang, B.-H., "Flexible speech understanding based on combined key-phrase detection and verification," *IEEE Trans. Speech and Audio Processing*, vol. 6, pp. 558–568, 1998.

[52] Kay, S. M., *Fundamentals of Statistical Signal Processing. Volume II: Detection Theory*. Prentice Hall, 1998.

[53] Kennedy, L. and Hauptmann, A., "LSCOM lexicon definition and annotation version 1.0," tech. rep., Columbia University, 2006.

[54] Kirchhoff, K., "Combining articulatory and acoustic information for speech recognition in noisy and reverberant environments," in *Proc. of ICSLP*, pp. 891–894, 1998.

[55] Koh, K., Kim, S.-J., and Boyd, S., "An interior-point method for large-scale l1-regularized logistic regression," *J. Mach. Learn. Res.*, vol. 8, pp. 1519–1555, 2007.

[56] Kraaij, W., Smeaton, A. F., Over, P., and Arlandis, J., "TRECVID 2004 - an overview," tech. rep., National Institute of Standards and Technology, 2005.

[57] Launay, B., Siohan, O., Surendran, A., and Lee, C.-H., "Towards knowledge-based features for HMM based large vocabulary automatic speech recognition," in *Proc. ICASSP*, 2002.

[58] Lee, C.-H., "On automatic speech recognition at the dawn of the 21st century," *IEICE Trans. INF. & SYST.*, vol. E86-D, pp. 377–396, March 2003.

[59] Lee, C.-H., "From knowledge-ignorant to knowledge-rich modeling: A new speech research paradigm for next generation automatic speech recognition," in *Proc. InterSpeech*, 2004.

[60] Lee, C.-H. and Huo, Q., "On adaptive decision rules and decision parameter adaptation for automatic speech recognition," *Proc. IEEE*, vol. 88, pp. 1241–1269, 2000.

[61] Leggetter, C. J. and Woodland, P. C., "Flexible speaker adaptation using maximum likelihood linear regression," in *Proc. ARPA Spoken Language Technology Workshop*, 1995.

[62] Leonard, R. G., "A database for speaker-independent digit recognition," in *Proc. of ICASSP*, 1984.

[63] Li, J., Tsao, Y., and Lee, C.-H., "A study on knowledge source integration for rescoring in automatic speech recognition," in *Proc. of ICASSP*, 2005.

[64] Li, X.-H., Chang, E., and Dai, B.-Q., "Improving speaker verification with figure of merit training," in *Proc. of ICASSP*, pp. 693–696, 2002.

[65] Lin, C.-Y., Tseng, B. L., and Smith, J. R., "VideoAnnEx: IBM MPEG-7 annotation tool," in *Proc. of ICME*, 2003.

[66] Lin, C.-J., Weng, R. C., and Keerthi, S. S., "Trust region Newton method for logistic regression," *Journal of Machine Learning Research*, vol. 9, pp. 627–650, 2008.

[67] Liu, S. A., "Landmark detection for distinctive feature based speech recognition," *JASA*, pp. 3417–3430, 1996.

[68] Ma, C., Byun, B., Kim, I., and Lee, C.-H., "A detection-based approach to broadcast news video story segmentation," in *Proc. of ICASSP*, 2009.

[69] Ma, C. and Chang, E., "Comparison of discriminative training methods for speaker verification," in *Proc. of ICASSP*, 2003.

[70] Ma, C. and Lee, C.-H., "Speaker verification based on combining speaker individuality parameter selection and decision," in *Proc. of ASRU*, 2005.

[71] Ma, C. and Lee, C.-H., "A study on word detector design and knowledge-based pruning and rescoring," in *Proc. of InterSpeech*, 2007.

[72] Ma, C. and Lee, C.-H., "Unsupervised anchor shot detection using multimodal spectral clustering," in *Proc. of ICASSP*, 2008.

[73] Ma, C., Nguyen, P., and Mahajan, M., "Finding speaker identities with a conditional maximum entropy model," in *Proc. of ICASSP*, pp. 261–264, 2007.

[74] Ma, C., Tsao, Y., and Lee, C.-H., "A study on detection based automatic speech recognition," in *Proc. of InterSpeech*, 2006.

[75] Ma, C. and Lee, C.-H., "An efficient gradient computation approach to discriminative fusion optimization in semantic concept detection," in *Proc. of ICPR*, 2008.

[76] Manning, C. D., Raghavan, P., and Schütze, H., *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[77] Martin, A., Doddington, G., Kamm, T., Ordowski, M., and Przybocki, M., "The DET curve in assessment of detection task performance," in *Procedings of EuroSpeech*, pp. 1895–1898, 1997.

[78] Mermelstein, P., "Distance measures for speech recognition, psychological and instrumental," *Pattern Recognition and Artificial Intelligence*, pp. 374–388, 1976.

[79] Miller, G. A. and Nicely, P., "An analysis of perceptual confusions among some english consonants," *Journal of the Acoustical Society of America*, vol. 27, p. 338352, 1955.

[80] Morgan, N., Chen, B., Zhu, Q., and Stolcke, A., "Trapping conversational speech: Extending TRAP/tandem approaches to conversational telephone speech recognition," in *Proc. ICASSP*, 2004.

[81] Neyman, J. and Pearson, E. S., "On the problem of the most efficient tests of statistical hypotheses," *Philosophical Transactions of the Royal Society of London*, vol. 231, pp. 289–337, 1933.

[82] Ng, A., Jordan, M., and Weiss, Y., "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, 2001.

[83] Nigam, K., Mccallum, A., Thrun, S., and Mitchell, T., "Text classification from labeled and unlabeled documents using EM," in *Machine Learning*, pp. 103–134, 1999.

[84] Nocedal, J. and Wright, S. J., *Numerical Optimization*. Springer, 2 ed., 2006.

[85] Povey, D. and Woodland, P., "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. of ICASSP*, 2002.

[86] Quenot, G., Petersohn, C., Over, P., and Walker, K., *TRECVID 2003 Keyframes & Transcripts*. LDC, 2007.

[87] Rabiner, L. R., "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–286, 1989.

[88] Rabiner, L. and Juang, B.-H., *Fundamentals of Speech Recognition*. Prentice Hall PTR, 1993.

[89] Rokui, J., "Rapid discriminative learning based on misclassification measure," *Syst. Comput. Japan*, vol. 37, pp. 58–68, 2006.

[90] Rose, K., "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," in *Proceedings of the IEEE*, pp. 2210–2239, 1998.

[91] Rosenfeld, R., *Adaptive statistical language modeling: a maximum entropy approach*. PhD thesis, Carnegie Mellon University, 1994.

[92] Santo, M. D., Foggia, P., Percannella, G., Sansone, C., and Vento, M., "An unsupervised algorithm for anchor shot detection," in *Proc. of ICPR*, 2006.

[93] Schapire, R. E., "The strength of weak learnability," *Machine Learning*, vol. 5, pp. 197–227, 1990.

[94] Schlüter, R., Macherey, W., Müller, B., and Ney, H., "Comparison of discriminative training criteria and optimization methods for speech recognition," *Speech Communication*, vol. 34, pp. 287–310, 2001.

[95] Sindhwani, V. and Keerthi, S. S., "Large scale semi-supervised linear SVMs," in *Proceedings of ACM SIGIR*, pp. 477–484, 2006.

[96] Siniscalchi, S. M., Li, J., and Lee, C.-H., "A study on lattice rescoring with knowledge scores for automatic speech recognition," in *Proc. of InterSpeech*, 2006.

[97] Siniscalchi, S. M., Svendsen, T., and Lee, C.-H., "A phonetic feature based lattice rescoring approach to LVCSR," in *Proc. ICASSP*, 2009.

[98] Smeaton, A. F., Kraaij, W., and Over, P., "TRECVID 2003 - an overview," tech. rep., National Institute of Standards and Technology, 2004.

[99] Tibshirani, R., "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

[100] Tikhonov, A. N. and Arsenin, V. A., *Solution of Ill-posed Problems*. Winston & Sons, Washington, 1977.

[101] Turk, M. and Pentland, A., "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, pp. 71–86, 1991.

[102] Viola, P. and Jones, M., "Rapid object detection using a boosted cascade of simple features," in *Proc. of CVPR*, 2001.

[103] Wald, A., *Statistical Decision Functions*. John Wiley and Sons, 1950.

[104] Warren, R. M. and Warren, R. P., "Auditory illusions and confusions," *Scientific American*, vol. 223, pp. 30–36, 1970.

[105] Y. Yue, T. Finley, F. R. and Joachims, T., "A support vector method for optimizing average precision," in *Proc. of SIGIR*, 2007.

[106] Yanagawa, A., Hsu, W., and Chang, S.-F., "Anchor shot detection in TRECVID-2005 broadcast news videos," tech. rep., Columbia University, 2005.

[107] Zhang, H., Gong, Y., Smoliar, S. W., and Tan, S. Y., "Automatic parsing of news video," in *Proc. of ICMCS*, 1994.

[108] Zhu, X., "Semi-supervised learning literature survey," Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[109] Zucker, S. W., "Computer vision and human perception: An essay on the discovery of constraints," in *Proc. of IJCAI*, 1981.