DISS. ETH NO.

Towards Robust Audio-Visual Speech Recognition

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by
TOFIGH NAGHIBI
MSc, Sharif University of Technology, Iran
born June 8, 1983
citizen of Iran

accepted on the recommendation of
, examiner
, co-examiner

2015

Contents

List of Abbreviations

AAM	active appearance model
A-ASR	audio-based automatic speech recognition
ADABoost	adaptive Boosting
ANN	artificial neural network
ASM	active shape model
ASR	automatic speech recognition
AUC	area under the curve
A-VAD	audio-based voice activity detection
AV-ASR	audio-visual automatic speech recognition
BE	backward elimination
BoW	bag of words
CART	classification and regression tree
CD-MABoost	combining dataset with MABoost
COBRA	convex based relaxation approximation
DCT	discrete cosine transform
fps	frame rate per second
FS	forward selection
GMM	Gaussian mixture model
HMM	hidden Markov model
ISCN	invariant scattering convolution network
ISMA	ISCN and MABoost posterior probabilities features

JMI	joint mutual information
KL	Kullback-Leibler
LBP	local binary pattern
LDA	linear discriminant analysis
MABoost	mirror ascent boosting
MADABoost	modified adaptive boosting
MAPP	Mu-MABoost posterior probabilities
MFCC	mel-frequency cepstral coefficient
MIFS	mutual information feature selection
ML	machine learning
MRI	magnetic resonance imaging
mRMR	minimal redundancy maximal relevance
Mu-MABoost	multiclass-MABoost
MVDR	minimum variance distortionless response
PAC	probably approximately correct
PCA	principle component analysis
RF	random forest
ROC	receiver operating characteristic
SDP	semidefinite programming
SIFT	scale-invariant feature transform
SSP	subset selection problem
SVM	support vector machine
TF	transfer function
VAD	voice activity detection
V-ASR	visual-based automatic speech recognition
V-VAD	visual-based voice activity detection

Abstract

Human speech production is a multimodal process, by nature. While usually the acoustic speech signals constitute the primary cue in human speech perception, visual signals can also substantially contribute, particularly in noisy environments. Research in automatic audio-visual speech recognition is motivated by the expectation that automatic speech recognition systems can also exploit the multimodal nature of speech. This thesis aims to explore the main challenges faced in realization and development of robust audio-visual automatic speech recognition (AV-ASR), i.e., developing (I) a high-accuracy speaker-independent lipreading system and (II) an optimal audio-visual fusion strategy.

Extracting a set of informative visual features is the first step to build an accurate visual speech recognizer. In this thesis, various visual feature extraction methods are employed. Some of these methods, however, encode visual information into very high-dimensional feature vectors. In order to reduce the computational complexity and the risk of overfitting, a new feature selection algorithm is introduced that selects a subset of informative visual features from the high-dimensional feature vector space. This feature selection algorithm considers mutual information between features and class labels (phonemes) to be the criterion and employs a semi-definite programming based search strategy for subset selection. The performance of the feature selection algorithm is analyzed and it is shown that the difference between the score of the selected feature subset and that of the optimal feature subset is bounded. That is, it guarantees that the score of the selected features is close to that of the optimal solution.

To achieve a speaker-independent visual speech recognizer, this thesis proposes to employ a pool of scale-invariant feature transform (SIFT) coefficients extracted from multiple color spaces. The ensemble of decision

tree classifiers trained with these features yields a high level of robustness against inter-speaker and illumination variations. While for voice activity detection two-dimensional SIFT features are used to build a statistical model, for lipreading we use three-dimensional SIFT features that can capture the time dynamics of video data. It is shown that using three-dimensional SIFT features gives a substantial recognition accuracy improvement in comparison with conventional visual features. The proposed AV-ASR achieves 70.5% utterance classification accuracy which is the highest accuracy reported for the Oulu dataset, in the speaker-independent setting.

While many boosting algorithms have been developed to train an ensemble of classifiers, most of them cannot be naturally generalized to the multiclass classification setting, which is a requirement in our application. In this thesis, we propose a framework to design boosting algorithms. This framework has the advantage that it can be naturally generalized to the multiclass setting. Several properties such as convergence rates and generalization error of this framework are analyzed. Moreover, multiple practically and theoretically interesting algorithms such as SparseBoost are derived. We show that the SparseBoost algorithm only uses a percentage of training samples at each training round (about half of the samples) while still converging to the optimal hypothesis in the sense of probably approximately correct (PAC) learning.

A common practice to fuse audio and visual information is to assign a reliability weight to each modality. It is shown in this thesis that a more suitable criterion to estimate the reliability weights is to maximize the area under a receiver operating characteristic curve (AUC) rather than frequently used criteria such as the recognition accuracy. Moreover, here we estimate a reliability weight for each feature. This generalizes the (conventional) two-dimensional stream weight estimation problem to a fairly high-dimensional problem. In order to efficiently estimate the reliability weights, we use a smoothed AUC function and adopt a variant of the projected gradient descent algorithm to maximize the AUC criterion in an online manner.

Audio-visual voice activity detection (AV-VAD) is an important prerequisite in many audio-visual applications. We propose a robust audio-visual voice activity detector which can be trained in a semi-supervised manner. This interesting property can be achieved by noting the fact that both audio and visual signals represent the same underlying event, namely speech production. In this approach, training data is labeled by iteratively training audio- and visual-based speech detectors and re-labeling the data in order to use it in the next round. The labeled data from the last iteration is then used to train

the final audio-visual voice activity detector. The proposed AV-VAD algorithm results in almost 96% frame-based detection rate (visual-based VAD yields 78% detection rate) on the GRID dataset in the speaker-independent setting.

Chapter 1

Introduction

1.1 Problem Statement

Speech, as the main communication medium among humans, is multimodal in nature. We all know from everyday life experience that observing the face of the person who talks to us usually improves our speech perception. This improvement is even more distinct when audio signal is degraded. For example, in a cocktail party situation where many people are talking at once, if we track the voice of a person whose face is not observable, our auditory system can only benefit from binaural processing where the desired sound source is localized by using the time and level differences between the signals received by ears. However, when we are able to observe the talker's face and gestures, a more complex phenomenon occurs. Recent studies have shown that focusing on the face of a talker in a crowded space enhances cortical selectivity in auditory cortex, which in turn results in diminishing the neural responses stimulated by the competing auditory input streams [GCSP13]. From a neurological standpoint, the mechanism by which visual inputs lead to elicit larger neural responses in auditory cortex is still largely unknown, the effect, however, is clear: when visual cues are available, noisy speech gets more intelligible.

Motivated by this observation and considering that there has been a significant improvement in audio-based automatic speech recognition (A-ASR)

[HDY+12, MHL+07, SS06, JHL97] over the last two decades, many researchers attempted to reduce the gap between human speech recognition performance and the performance of A-ASR in real world applications by integrating visual speech information and auditory speech information. This, however, turned out to be non-trivial. The first attempts in this direction [PNLM04, and references there in] clearly showed that appearance based visual features (pixel values and their affine transformations) provide very limited learnable speech information. Moreover, such features are highly speaker dependent and strongly affected by changing illumination conditions. That is, the commonly used visual features are not reliable. This raised several theoretical and practical questions.

The first and foremost question is which are the appropriate visual features? It is clear that due to the data processing theorem, features obtained by applying non-linear transformations over an image do not convey more information than the raw pixel values themselves. With some of these non-linearly transformed features however, a given learning algorithm can more efficiently search through the hypothesis space, which in turn may result in returning a more accurate model to represent visual speech. Thus, it is only meaningful to explore the goodness of a feature set with respect to a given learning algorithm. The follow up question is then, which is the appropriate learning algorithm for visual speech recognition?

The next challenging question in audio-visual automatic speech recognition (AV-ASR) is how to optimally combine audio and visual speech information. From a purely theoretical standpoint, if audio and visual feature streams were class conditionally independent, multiplying their likelihood functions and the a priori probability would result in optimal fusion (Bayes fusion). In reality however, our estimation of likelihood functions (particularly likelihood function of the video stream) may severely deviate from the true likelihood functions mainly due to mismatch between the distributions of training and test data and due to model inaccuracy. Therefore, a more complex fusion strategy is needed to weight the likelihood functions with respect to the reliability of the models.

This thesis will concentrate on both theoretical and practical aspects of feature selection and learning algorithms and their applications in AV-ASR in order to give answers that differ form conventional solutions to the above questions.

1.2 Scientific Contributions

The following contributions result from this thesis:

1. We propose a mutual information based feature selection algorithm and derive its approximation ratio which can be interpreted as a lower bound of the goodness of selected feature sets. To the best of our knowledge, this is the first (mutual information based) feature selection method with performance guarantee[1].

2. Employing the duality between linear online learning (hedging) and boosting, we derive a framework to design and develop strong classifiers. We show the algorithms derived from this framework are boosting algorithms in the probably approximately correct (PAC) sense, i.e., they drive the classification error to zero. Using this framework, we show that the MADABoost algorithm proposed in [DW00] is in fact a boosting algorithm. This is the first proof of its boosting property. Moreover, a sparse boosting algorithm is proposed which uses only a percentage of the training data in each training round resulting in a substantial memory and computational complexity reduction of learning process.

3. Most commonly used visual features for speech recognition are very sensitive to illumination variations and moreover are highly speaker dependent. To cope with speaker dependence issue, we propose to use 3 dimensional scale invariant feature transform (3D-SIFT). To improve illumination invariance property, we suggest to use multiple 3D-SIFT feature sets where features sets are extracted from different color spaces (obtained by nonlinear transformation of RGB color space) with some invariance properties. The final classifier is the ensemble of classifiers each trained with one of these feature sets. The resulting classifier yields high accuracy in speaker-independent mode and is robust against changing lighting conditions.

4. Using an algorithm derived from our proposed boosting framework (MABoost) in co-training setting discussed in [BM98], we develop a speaker independent audio-visual voice activity detection system that

[1] This performance guarantee is a non-zero (thus, non-trivial) lower bound of the ratio between the attained solution and the optimal solution of the underlying NP hard maximization problem.

can be trained in a semi-supervised manner. It is shown to be very reliable in different lighting conditions and for various speakers. Further, since it does (almost) not require labeled data to train, its intelligence (accuracy) improves as it is used, meaning that it can adapt to new users in an unsupervised way.

5. Many audio-visual fusion techniques proposed in the literature, optimize the fusion parameters with respect to the recognition accuracy on training data which is based on an implicit assumption that training conditions are sufficiently similar to test conditions. This, however, is a paradoxical assumption since if training and test conditions were similar enough, we could simply use Bayes fusion. To address this problem, we propose a multi-stream fusion scheme adopt the area under the receiver operating characteristic curve (AUC) as the design criterion. This algorithm can be trained in an online manner when the training set is large or parameter adaptation is required.

1.3 Structure of this Thesis

This thesis includes two parts: A more theoretical part where a feature selection algorithm and a boosting framework are described and a more practical part where these algorithms are employed to construct a robust audio-visual speech recognition system.

Chapter 2 describes different visual features employed in this thesis including SIFT, 3D-SIFT and ISCN. Moreover, the audio-visual datasets used for evaluations throughout this thesis are also presented in this chapter.

Chapter 3 introduces mutual information based feature selection algorithms. Particularly for COBRA, the semi-definite programming based feature selection, the approximation ratio is explored. Finally, the practical useability of this feature selection method is demonstrated by means of a relatively comprehensive experiments with 10 different datasets.

Chapter 4 introduces a boosting framework called MABoost. Several boosting algorithms derived from this framework, including SparseBoost, MADABoost and Mu-MABoost are presented and explored in this chapter.

Chapter 5 discusses in details the supervised and semi-supervised algorithm employed for audio-visual voice activity detection.

Chapter 6 gives the description of a decision tree based lip reading system. It explains how 3D-SIFT features can be used to train a multiclass classifier in order to achieve a robust lip reading system.

Chapter 7 is devoted to the information fusion problem in audio-visual speech recognition systems. It describes a fusion scheme for AV-ASR systems.

Chapter 8 gives some practical evidence for the advantages of using multichannel audio and visual data for improving recognition accuracy in noisy, reverberant environments.

Chapter 9 provides some final remarks and insights for future works.

Chapter 2

Visual Features and AV Datasets

According to Castleman's definition [Cas79], a feature is a function of one or more measurements, computed so that it quantifies some signification characteristics of an object. Despite the fact that over the last years various types of visual features have been proposed for V-ASR systems (see [ZZHP14, PNLM04]), none of them have gained wide acceptance (such as MFCCs in A-ASR systems) in real world deployment. Different visual feature extraction methods may roughly be categorized into four classes.

1. Appearance or texture based features including PCA, LDA, H-LDA etc. The main underlying concept among texture based features is that *all pixels encode visual speech information*. After extracting a region of interest (ROI) which usually contains mouth, lips and jaw, these methods apply traditional dimensionality reduction techniques (such as DCT, PCA or LDA) directly to the ROI to calculate the final feature sets. However, in these features the information relevant to visual speech is strikingly dominated by irrelevant information (noise for our goal) such as facial characteristics and skin color of the speaker. Furthermore, these features are very sensitive to illumination variations.

2. Shape based algorithms [Che01, MCB+02, CET01] such as AAM and ASM which consist of models for the visible speech articulatory parts.

These models, however, need a tedious training (facial points have to be labeled) and may not capture all relevant information due to the limitations of models (shapes). More importantly, they are highly sensitive to image quality (low resolution).

3. Motion based methods which mainly use optic flow features. Although using optic flow features alone results in poor performance, it was shown that they may be useful as complimentary features in AV-ASR systems [Gur09].

4. Invariant features which show different invariance properties with respect to illumination conditions, scaling, skin color, etc. Invariant features are achieved by applying non-linear transformations to an ROI and can be extracted either by computing features for each frame independently, i.e., spatial features such as SIFT and ISCN, or by considering the video data a three dimensional time series and computing informative spatio-temporal features for a video sequence.

   Our proposed feature sets may also be considered to go along this line with an additional bags-of-words (BoW) step resulting in a sparse feature vector (suitable for decision tree classifiers used as weak classifiers in our boosting algorithm). These features turn out to be highly speaker- and illumination-invariant.

While the last class of features are widely used in various machine vision and video classification applications [SAS07], they were rarely applied to the lipreading problem. In the first part of this chapter various invariant features that are later used to develop speaker-independent voice activity detection and speech recognition systems, are introduced. As shown in Chapter 6, using these features significantly improves the robustness of the V-ASR systems.

In the second part of this chapter, the datasets used in this thesis are explained and some samples from them are illustrated.

2.1 Visual Feature Descriptions

Two kinds of features can be extracted from an image: (I) *Global features* which can be seen as a function of an entire image. PCA and LDA features are some examples of global features. In this type of representation, each feature is a function of both foreground (interpreted as information we are

interested in) and background (interpreted as noise). (II) *Local features* which describe the characteristics of small regions in neighborhood of key points in an image. The key points, however, may belong to foreground or background. Thus, unlike global features, local features are functions of either foreground or background but not both at the same time. Since in lipreading, among many details and information in a face image, we are only interested in lip shape and location, it is advantageous to use local features. It is then the classifier's task to intelligently extracts information from foreground features and ignore the background features. In this thesis, only local features are employed due to the fact that these features provide robustness required by lipreading systems.

2.1.1 Scale-invariant Feature Transform (SIFT)

Scale-invariant feature transform (SIFT) descriptors are local features introduced by Lowe [Low04]. SIFT descriptors are one of the most widely used local features in machine-vision applications due to their invariance to scale, rotation, illumination changes, and viewing directions. The success of SIFT features can be attributed to two factors: First, it does not blindly use all the pixels. It finds local structures such as corners or blobs that are present in different views and different scales of an image and uses them as key points. Second, once the key points were detected, it provides descriptions for these key points which are partially invariant to translation, rotation, scaling, illumination and affine transformations. These descriptors will be the SIFT feature vectors representing an image.

SIFT descriptors are computed as follows: First, the image gradients at the sample points in a 2D neighborhood around a key point location are computed. Lowe argued that it is better to use the distribution of the gradient orientations rather than their raw values since this distribution is highly invariant to partial variations (such as scaling, rotation and translation). Thus, at the second stage, the histograms of orientations are computed. When computing the orientation histogram, the increments are weighted by the gradient magnitude and also weighted by a Gaussian window function centered at the key point. These orientation histograms measure how strong the gradient is in each direction. By forming multiple such histograms (according to Lowe method, 4 such histograms for 4 subregions around a key point) and concatenating them the SIFT descriptor for a key point is obtained. This vector is usually normalized by the $\ell 1$ or $\ell 2$ norm of the vector in order to achieve invariance to illumination changes.

In this thesis, SIFT features are used to train a visual voice activity detector (VAD). As shown in Chapter 5, invariance properties of SIFT features can significantly contribute to the robustness of the system.

2.1.2 Invariant Scattering Convolution Networks

Invariant scattering convolution network (ISCN) coefficients are wavelet based features computed by a network of wavelets applied to an image. Following the explosive success of convolutional neural networks, Bruna and Mallat introduced ISCN features [BM12] in order to extract higher order statistics of an image via a deep network of wavelets. In this representation, an image is first segmented into many overlapped subregions. The first layer of the network then outputs local features by (I) applying wavelet transforms to subregions and (II) removing their phases to obtain translation invariant property. Averaging the features of each subregion yields a feature set which is both deformation-invariant (due to the averaging) and translation-invariant. As shown in [BM12], these features turn out to be SIFT-type descriptors. The next layers, however, provide higher-order statistics by applying finer wavelets in various directions[1] to the outputs of the previous layer.

These higher-order features show strong discriminative power, particularly when the underlying distribution of the data is highly non-Gaussian. This, however, comes at the expense of exponentially expanding the feature space due to the fact that the number of ISCN features exponentially increases with the number of layers and the number of directions along which the wavelet transform is computed.

In our experiments, for instance, a 3-layer scattering convolution network with an 8-directional Morlet wavelet with the maximum spatial resolution 8, yields 55552 ISCN features for an ROI of size 128×128. This representation can be compressed by applying DCT transform on ISCN features and only selecting the 20% of the DCT features with the highest energy, which in our experiments were about 10000 features. Since, given the limited computational, memory and data resources, with this number of features it is still difficult to devise a reliable statistical model, a feature selection algorithm is developed to select a small subset of informative ISCN features used in training HMM-GMM models. This feature selection method is the topic of the next chapter.

[1] In 2D frequency space, directional wavelets can be described by rotating and scaling the base wavelet function.

2.1.3 3D-SIFT

There is a growing body of research attempting to generalize the successful 2D descriptors to 3D descriptors in order to take the third dimension of video data (time in video and Z in MRI images) into account. Several 3D feature sets proposed in the literature include 3D-SIFT [SAS07], HoG3D [KMS08], spatio-temporal BoW [NWF08] and LBP-TOP [ZBP09]. HoG3D and 3D-SIFT, however, are based on histograms of gradient orientations and can be seen as the direct generalization of the popular SIFT descriptors. Due to their compliance with the 3D nature of video data, they achieved a great deal of success in some applications such as action recognition.

Local descriptors are used to represent a local fragment of a video. Usually informative points (key points) of a video are selected either by a key point detector (e.g., Harris-Laplace detector [MS02]) or by dense sampling. Since we only extract features from a small ROI (lip and mouth region), dense sampling is employed in this thesis. As in SIFT, after selecting the key points, the next step is to compute the gradient magnitudes and orientations at the sample points in a 3D neighborhood around the key point location. Given a key point $\mathbf{s} = (x_s, y_s, t_s)$, a 3D neighborhood is considered to be a cuboid volume with its center of gravity lying at the key point location. This 3D cuboid is described by $C = (\mathbf{s}, w, l, h)$ where w, h and l are its width (along X axis), height (along Y axis) and height (along time axis), respectively. The gradient magnitudes are weighted by a Gaussian window which in our case is a sphere centered at the key point location. The 3D cuboid is first divided into 2x2x2 sub-volumes. The gradient orientations in each sub-volume are then accumulated into a sub-histogram with 80 bins, where 80 is the number of polyhedron faces used to tessellate the sphere (in order to quantize the gradient orientations). A sub-histogram represents the visual information of its corresponding sub-volume. All the sub-histograms are concatenated then to construct the final feature vector of length 640.

2.1.4 Bag-of-Words

The bag-of-words (BoW) model is one of the most popular feature representation in machine-vision and visual classification tasks. The key idea in BoW is in fact borrowed from text mining. Usually, the standard steps in text retrieval systems to extract features from a document is: (I) Parsing a document

into words (II) Assigning a unique identifier (code) to each word (III) representing the document by a vector with components given by the frequency of occurrence of the words. Following this line of thought, Sivic and Zisserman introduced the BoW model for object retrieval and visual search in videos [SZ03]. The BoW method can be outlined as follows:

1. Extracting raw features from an image or a video. In this thesis, we use SIFT and 3D-SIFT as the raw features for representing the ROIs.

2. Constructing a codebook with K codes by employing a clustering approach such as K-means and assigning an index $i \in \{1, \ldots, K\}$ to each feature vector according to its cluster membership.

3. Representing each ROI by a K-dimensional vector where the i-th element is the frequency of occurrence of the index i in the given ROI.

Two important problems can be handled by using BoW in classification.

First, in lipreading, ROIs may have different sizes (because of the variations of speakers' lip shapes and sizes). Due to the changes in image sizes, we may extract different number of feature vectors from different ROIs. It is then not clear how to represent ROIs with equal-size feature vectors. One remedy is to artificially resize all images to have an equal size. This idea is used in Chapter 3 to train a visual speech recognizer with ISCN features. Another solution is to use the BoW model which sidesteps this problem by assigning equal-size feature vectors to ROIs.

Second, features generated by the BoW method are highly robust against, small variations and largely unaffected by a change in camera viewpoint, the object's scale and scene illumination. This robustness is the direct result of two factors: First, employing vector quantization and second, using the distribution (frequency of occurrence) of the raw features as the final representation of an image. Both of these factors reduce the amount of noise in raw features and consequently, increase the robustness of the system.

2.2 Datasets

Compared with audio-only dataset, the number of audio-visual datasets suitable for training an AV-ASR is much smaller and even fewer of them are freely available. In this thesis, we used four datasets which were commonly

used in the relevant literature: CUAVE [PGTG02], GRID [CBCS06], Oulu [ZBP09] and AVletter [MCB+02]. Moreover, we also recorded a multi-channel audio-visual dataset which is used in the last chapter to conclude this thesis.

Table 2.1 summarizes some of the important properties of the datasets used in this thesis. Note that the GRID dataset originally contains 32 speakers while we only used 16 of them in our evaluations.

Dataset Name	Resolution	# Audio channels	Light cond.	Speakers
CUAVE	720x480	1	controlled	36
Oulu	720x576	1	controlled	20
GRID	720x576	1	controlled	16
AVLetter	unknown	0	controlled	10
ETHDigits	640x480	8	time-varying	15

Table 2.1: Summery of datasets utilized in this thesis.

2.2.1 CUAVE

The first dataset used in this thesis is CUAVE [PGTG02], which contains the digits from zero to nine repeated five times by 36 speakers. This dataset offers reasonably large speaker variability which is necessary to train a robust recognizer. The dataset has been recorded at a resolution of 720x480 with the NTSC standard of 29.97 fps [PGTG02]. We, however, do not directly work with the raw data. Due to the effort of Mihai Gurban in his PhD work [Gur09],

Figure 2.1: Manually centered, rotated and scaled ROIs extracted from CUAVE dataset.

the preprocessed data, where the video data is interlaced, mouth regions are manually cropped, centered and rotated to be horizontal, is available for this dataset. Four region of interests (ROIs) from this preprocessed dataset are depicted in Figure 2.1.

Each ROI is a square image with 128x128 pixels. All ROIs are of equal size and the feature vectors representing these ROIs are also of equal size. Naturally, working with manually detected ROIs leads to overly optimistic results. However, since our focuses in Chapters 3 and 7, where the CUAVE dataset is employed, are on feature set evaluation and audio-visual information fusion, this issue can be ignored. Later in Chapter 6, when we opt the GRID dataset, which is much larger than CUAVE and for which no manually detected ROIs are available, the effect of the automatic ROI detection is explored.

2.2.2 GRID

GRID corpus has been introduced in [CBCS06]. It consists of high-quality audio and video recordings of 1000 utterances spoken by each of 34 talkers. Sentences are simple, syntactically identical phrases such as “place green at B 4 now”. The sentence structure for the Grid corpus is indicated in Table 2.2.

Each sentence consists of six words. Audio-visual recordings were made in a single-walled acoustically isolated booth. Each recording lasts exactly 3 second and consists of 75 frames, i.e., 25 frames per second. The image resolution is 720x576 pixels. The sampling frequency of the audio data is 44.1 kHz, which in our experiments was down sampled to 8 kHz. Some sample frames of this corpus are shown in Figure 2.2.

Command	**Color***	**Preposition**	**Letter***	**Digit***	**Adverb**
bin	blue	at	A–Z	0–9	again
lay	green	by	excluding W		now
place	red	in			please
set	white	with			soon

Table 2.2: Sentence structure for the GRID corpus. The keywords are shown by asterisks. Each sentence consists of six words, three of which are keywords.

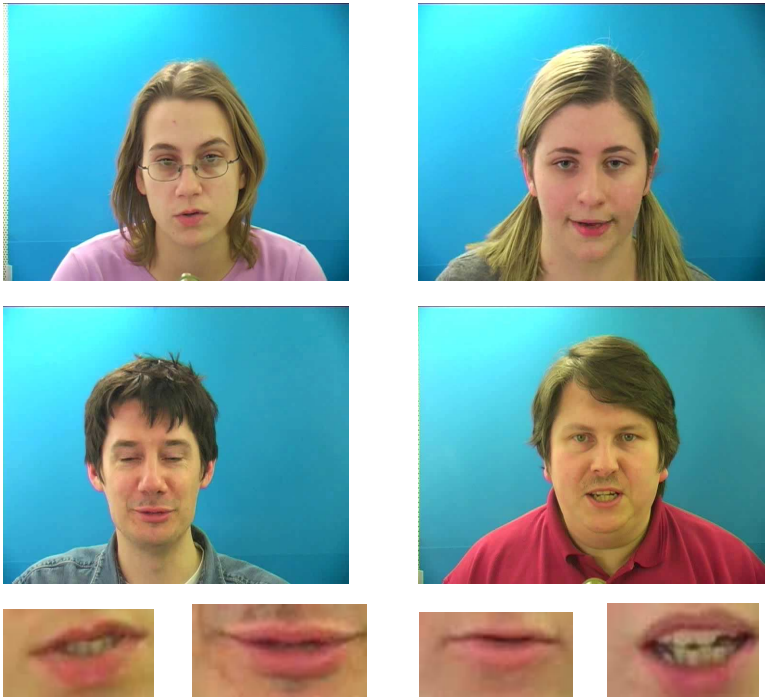

Figure 2.2: Top four images are sample frames from the GRID corpus and the bottom four images are automatically extracted mouth regions. Different ROIs may have different sizes.

Unlike the CUAVE dataset, mouth regions are not manually labeled. In order to extract ROIs, in this thesis we used a fully automatic facial point detection algorithm proposed by Dantone et al. [DGFVG12]. It is a real-time facial point detection approach based on the conditional regression forest and yields high accuracy in most cases. The detected ROIs may have different sizes due to the various mouth shapes, speakers's facial characteristics, their distance to the camera and detection error. Four of these ROIs are illustrated in Figure 2.2.

The GRID dataset is more than 40 GB. Hence, we only used a subset of the corpus in our experiments by randomly selecting 400 sentences from each of the first 16 speakers in the corpus. This reduces the memory size for processing and makes it possible to run our learning algorithms in the batch learning mode.

2.2.3 Oulu

The Oulu dataset collected by Zhao et al. [ZBP09] is a small corpus containing 10 phrases listed in Table 2.3. The video data is recorded by a SONY DSR-200AP 3CCD-camera with a frame rate of 25 fps. The image resolution is 720x576 pixels. This dataset includes 20 persons, each uttering the ten phrases listed in Table 2.3 for five times. Audio files are single channel 16 bit per sample uncompressed wave files with sampling frequency 48 kHz. With this dataset, semi-automatically cropped ROIs are also provided. These mouth regions were determined by giving eyes positions manually in each frame to an automatic mouth detection system. Figure 2.3 depicts four frames and four extracted ROI of this dataset. While the video data is colored, the provided ROIs with this dataset are gray scale images. Therefore, as in the GRID dataset, we automatically extracted our own ROIs, since as shown in Chapter 6, colored images are more suitable for lipreading due to valuable information in colors, particularly the red color of lips. By Transforming RGB

C1	**C2**	**C3**	**C4**	**C5**
Excuse me	Good bye	Hello	How are you	Nice to meet you
C6	**C7**	**C8**	**C9**	**C10**
See you	I am sorry	Thank you	Have a good time	You are welcome

Table 2.3: Ten phrases used in the Oulu dataset.

images to multiple color spaces, we improve the robustness of the system and increase the convergence speed of the training phase.

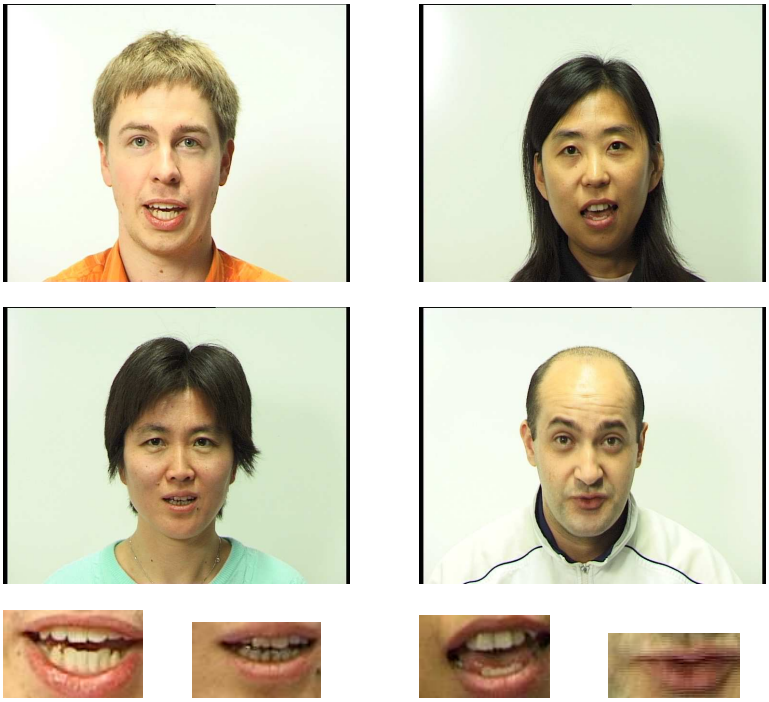

Figure 2.3: Top four images are sample frames from the Oulu corpus and the bottom four images are automatically extracted mouth regions.

2.2.4 AVletter

The AVletters dataset [MCB+02] consists of 10 speakers saying the letters A to Z, three times each. The dataset only provided pre-extracted lip regions at 60x80 pixels in gray scale. As there is no raw audio data available for this dataset, we only used it for evaluation of lipreading systems. Figure 2.4 demonstrates four example ROIs of this dataset.

Figure 2.4: Four ROI examples from AVletter dataset which consists repetitions of letters A to Z.

2.2.5 ETHDigits

Since all the explained datasets were collected under controlled conditions, their high quality audio-visual data are not representative of real-world examples. Therefore, we set up a recording system to collect our own audio-visual dataset.

Figure 2.5: Eight channel microphone array used for audio acquisition in the ETHDigits dataset.

ETHDigits is an audio-visual dataset recorded in a highly reverberant office room with more than 1 second reverberation time T_{60}. The audio data is captured by 8 microphones shown in Figure 2.5 and video signals are recorded by a Kinect for Xbox 360. In this work however, we only use the RGB camera of Kinect, which has 640x480 pixel resolution and operates at a frame rate of 30 Hz. Unlike the previous datasets, we did not record the visual data under controlled conditions. Depending on how many lights were on during a recording session, what time of day it was and whether the sun was shining, amount of light in the room may largely vary, which in turn, results in large variation of the visual data quality. The ETHDigits dataset consists of 15 speakers and each speaker repeats a sequence of numbers from one to ten for 5 times. Digits were presented on a computer screen located about 45 centimeters away from talkers and both Kinect and the microphone

array were mounted on top of this screen. Unlike the CUAVE database, here the five digit sequences appear with different speeds on the screen (the later the faster). Namely, there are longer silence durations between numbers of the first three sequences than that of the last two sequences.

Some samples of this dataset can be seen in Figure 2.6.

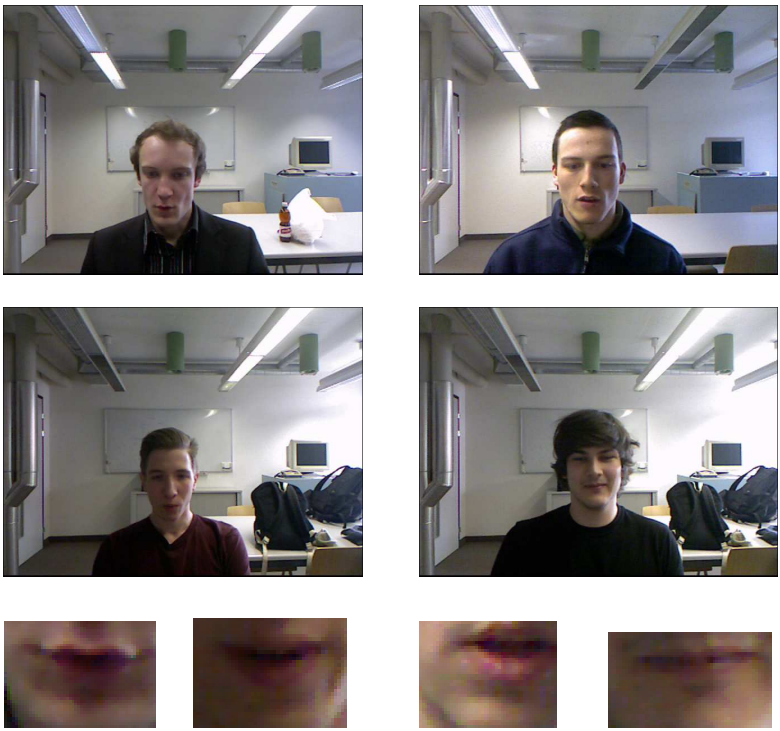

Figure 2.6: Top four images are sample frames from the ETHDigits dataset and the bottom four images are their corresponding automatically extracted mouth regions.

Chapter 3

Feature Selection

For various reasons, feature selection is a necessary preprocessing block in many machine learning applications. The main reason, however, is simply the lack of *enough* data. In this chapter, various aspects of the feature selection problem are investigated and a feature selection algorithm is developed which later is employed to select a small subset of informative features from ISCN coefficients in order to train a GMM-HMM based lipreading system.

The main two issues in feature subset selection are finding an appropriate measure function that can be fairly fast and robustly computed for high-dimensional data and a search strategy to optimize the measure over the subset space in a reasonable amount of time. In this chapter mutual information between features and class labels is considered to be the measure function. Two series expansions for mutual information are proposed, and it is shown that most heuristic criteria suggested in the literature are truncated approximations of these expansions.

As of search strategy we suggest a parallel search strategy based on semi-definite programming (SDP) that can search through the subset space in polynomial time. By exploiting the similarities between the proposed algorithm and an instance of the maximum-cut problem in graph theory, the approximation ratio of this algorithm is derived and is compared with the approximation ratio of the backward elimination method.

3.1 Introduction to Feature Selection Problem

From a purely theoretical point of view, given the underlying conditional probability distribution of a dependent variable C and a set of features $\mathbf{X}$, the Bayes decision rule can be applied to construct the optimum induction algorithm. However, in practice learning machines are not given access to this distribution, $Pr(C|\mathbf{X})$. Therefore, given a feature vector or variables $\mathbf{X} \in R^N$, the aim of most machine learning algorithms is to approximate this underlying distribution or estimate some of its characteristics. Unfortunately, in most practically relevant data mining applications, the dimensionality of the feature vector is quite high making it prohibitive to learn the underlying distribution. For instance, gene expression data or images may easily have more than tens of thousands of features. While, at least in theory, having more features should result in a more discriminative classifier, it is not the case in practice because of the computational burden and the overfitting effect.

High-dimensional data poses different challenges on induction and prediction algorithms. Essentially, the amount of data to sustain the spatial density of the underlying distribution increases exponentially with the dimensionality of the feature vector, or alternatively, the sparsity increases exponentially given a constant amount of data. Normally in real-world applications, a limited amount of data is available and obtaining a sufficiently good estimate of the underlying high-dimensional probability distribution is almost impossible unless for some special data structures or under some assumptions (independent features, etc).

Thus, dimensionality reduction techniques, particularly feature extraction and feature selection methods, have to be employed to reconcile idealistic learning algorithms with real-world applications.

3.1.1 Various Feature Selection Methods

In the context of feature selection, two main issues can be distinguished. The first one is to define an appropriate measure function to assign a score to a set of features. The second issue is to develop a search strategy that can find the optimal (in a sense of optimizing the value of the measure function) subset of features among all feasible subsets in a reasonable amount of time.

Different approaches to address these two problems can roughly be categorized into three groups: Wrapper methods, embedded methods and filter methods.

Wrapper methods [Koh96] use the performance of an induction algorithm (for instance a classifier) as the measure function. Given an inducer $\mathcal{I}$, wrapper approaches search through the space of all possible feature subsets and select the one that maximizes the induction accuracy. Most of the methods of this type require to check all the possible 2^N subsets of features and thus, may rapidly become prohibitive due to the so-called combinatorial explosion. Since the measure function is a machine learning (ML) algorithm, the selected feature subset is only optimal with respect to that particular algorithm, and may show poor generalization performance over other inducers.

The second group of feature selection methods are called embedded methods [NSS04] and are based on some internal parameters of the ML algorithm. Embedded approaches rank features during the training process and thus simultaneously determine both the optimal features and the parameters of the ML algorithm. Since using (accessing) the internal parameters may not be applicable in all ML algorithms, this approach cannot be seen as a general solution to the feature selection problem. In contrast to wrapper methods, embedded strategies do not require to run the exhaustive search over all subsets since they mostly evaluate each feature individually based on the score calculated from the internal parameters. However, similar to wrapper methods, embedded methods are dependent on the induction model and thus the selected subset is somehow tuned to a particular induction algorithm.

Filter methods, as the third group of selection algorithms, focus on filtering out irrelevant and redundant features in which irrelevancy is defined according to a predetermined measure function. Unlike the first two groups, filter methods do not incorporate the learning part and thus show better generalization power over a wider range of induction algorithms. They rely on finding an optimal feature subset through the optimization of a suitable measure function. Since the measure function is selected independently of the induction algorithm, this approach decouples the feature selection problem from the following ML algorithm.

The first contribution of this work is to analyze the popular mutual information measure in the context of the feature selection problem. The mutual information function is expanded in two different series and is shown that most of the previously suggested information-theoretic criteria are the first or second order truncation-approximations of these expansions. The first expansion is based on generalization of mutual information and has already appeared in the literature while the second one is new, to the best of our knowledge. The well-known minimal Redundancy Maximal Relevance (mRMR)

score function can be immediately concluded from the second expansion.

3.1.2 Feature Selection Search Strategies

Alternatively, feature selection methods can be categorized based on the search strategies they employ. Popular search approaches can be divided into four categories: Exhaustive search, greedy search, projection and heuristic. A trivial approach is to exhaustively search in the subset space as it is done in wrapper methods. However, as the number of features increases, it can rapidly become infeasible. Hence, many popular search approaches use greedy hill climbing, as an approximation to this NP-hard combinatorial problem. Greedy algorithms iteratively evaluate a candidate subset of features, then modify the subset and evaluate if the new subset is an improvement over the old one. This can be done in a forward selection setup which starts with an empty set and adds one feature at a time or with a backward elimination process which starts with the full set of features and removes one feature at each step. The third group of the search algorithms are based on targeted projection pursuit which is a linear mapping algorithm to pursue an optimum projection of data onto a low dimensional manifold that scores highly with respect to a measure function [FT74]. In heuristic methods, for instance genetic algorithms, the search is started with an initial subset of features which gradually evolves toward better solutions.

Recently, two convex quadratic programing based methods, QPFS in [RHEC10] and SOSS in [NHP13] have been suggested to address the search problem. QPFS is a deterministic algorithm and utilizes the Nyström method to approximate large matrices for efficiency purposes. SOSS on the other hand, has a randomized rounding step which injects a degree of randomness into the algorithm in order to generate more diverse feature sets.

Developing a new search strategy is another contribution of this chapter. Here, a new class of search algorithms is introduced which is based on semidefinite Programming (SDP) relaxation. The feature selection problem is reformulated as an instance of (0-1)-quadratic integer programming. This integer programming optimization is then relaxed to an SDP problem, which is convex and hence can be solved with efficient algorithms [BV04]. It is shown that it usually gives better solutions than greedy algorithms in the sense that its approximate solution is more probable to be closer to the optimal point of the criterion.

3.2 Mutual Information Pros and Cons

Let us consider an N dimensional feature vector $\mathbf{X} = [X_1, X_2, ..., X_N]$ and a dependent variable C which can be either a class label in case of classification or a target variable in case of regression. The mutual information function is defined as a distance from independence between $\mathbf{X}$ and C measured by the Kullback-Leibler divergence [CT91]. Basically, mutual information measures the amount of information shared between $\mathbf{X}$ and C by measuring their dependency level. Denote the joint pdf of $\mathbf{X}$ and C and its marginal distributions by $Pr(\mathbf{X}, C)$, $Pr(\mathbf{X})$ and $Pr(C)$, respectively. The mutual information between the feature vector and the class label can be defined as follows:

$$I(X_1, X_2, \ldots, X_N; C) = I(\mathbf{X}; C) = \int Pr(\mathbf{X}, C) \log \frac{Pr(\mathbf{X}, C)}{Pr(\mathbf{X}) Pr(C)} \, \mathrm{d}\mathbf{X} \, \mathrm{d}C \tag{3.1}$$

It reaches its maximum value when the dependent variable is perfectly described by the feature set. In this case mutual information is equal to $H(C)$, where $H(C)$ is the Shannon entropy of C.

Mutual information can also be considered a measure of set intersection [Rez61]. Namely, let $\mathbb{A}$ and $\mathbb{B}$ be event sets corresponding to random variables A and B, respectively. It is not difficult to verify that a function μ defined as:

$$\mu(\mathbb{A} \cap \mathbb{B}) = I(A; B) \tag{3.2}$$

satisfies all three properties of a formal measure over sets [Yeu91] [Bog07], i.e., non-negativity, assigning zero to empty set and countable additivity. However, as it is seen later, the generalization of the mutual information measure to more than two sets will no longer satisfy the *non-negativity* property and thus can be seen as a signed measure which is the generalization of the concept of measure by allowing it to have negative values.

There are at least three reasons for the popularity of the use of mutual information in feature selection algorithms.

1. Most of the suggested non information-theoretic score functions are not formal set measures (for instance correlation function). Therefore, they cannot assign a score to a set of features but rather to individual features. However, mutual information as a formal set measure is able to evaluate all possible informative interactions and complex functional relations between

features and as a result, represent the complete information contained in a set of features.

2. The relevance of the mutual information measure to misclassification error is supported by the existence of bounds relating the probability of misclassification of the Bayes classifier, P_e, to the mutual information. More specifically, Fano's weak lower bound [Fan61] on P_e,

$$1 + P_e \log_2(n_y - 1) \geq H(C) - I(\mathbf{X}; C) \tag{3.3}$$

where n_y is the number of classes and the Hellman-Raviv [HR70] upper bound,

$$P_e \leq \frac{1}{2}(H(C) - I(\mathbf{X}; C)) \tag{3.4}$$

on P_e, provide somewhat a performance guarantee.

As it can be seen in (3.3) and (3.4), maximizing the mutual information between $\mathbf{X}$ and C decreases both upper and lower bounds on misclassification error and guarantees the goodness of the selected feature set. However, there is somewhat of a misunderstanding of this fact in the literature. It is sometimes wrongly claimed that maximizing the mutual information results in minimizing the P_e of the optimal Bayes classifier. This is an unfounded claim since P_e is not a monotonic function of the mutual information. Namely, it is possible that a feature vector $\mathbf{A}$ with less relevant information-content about the class label C than a feature vector $\mathbf{B}$ yields a lower classification error rate than $\mathbf{B}$. The following example may clarify this point.

Example 3.1. Consider a binary classification problem with equal number of positive and negative training samples and two binary features X_1 and X_2. The goal is to select the optimum feature for the classification task. Suppose the first feature X_1 is positive if the outcome is positive. However, when the outcome is negative, X_1 can take both positive and negative values with the equal probability. Namely, $Pr(X_1{=}1|C{=}1) = 1$ and $Pr(X_1{=}-1|C{=}-1) = 0.5$. In the same manner, the likelihood of X_2 is defined as $Pr(X_2{=}1|C{=}1) = 0.9$ and $Pr(X_2{=}-1|C{=}-1) = 0.7$. Then, the Bayes classifier with feature X_1 yields the classification error:

$$\begin{aligned} P_{e1} =& Pr(C{=}{-}1)Pr(X_1{=}1|C{=}{-}1) \\ &+ Pr(C{=}1)Pr(X{=}{-}1|C{=}1) = 0.25 \end{aligned} \tag{3.5}$$

Similarly, the Bayes classifier with X_2 yields $P_{e1} = 0.2$ meaning that, X_2 is a better feature than X_1 in the sense of minimizing the probability of misclassification. However, unlike their error probabilities, $I(X_1; C) = 0.31$, is

greater than $I(X_2;C) = 0.29$. That is, X_1 conveys more information about the class label in the sense of Shannon mutual information than X_2.

A more detailed discussion can be found in [FDV12]. However, it is worthwhile to mention that although using mutual information may not necessarily result in the highest classification accuracy, it guarantees to reveal a salient feature subset by reducing the upper and lower bounds of P_e.

3. By adapting classification error as a criterion, most standard classification algorithms fail to correctly classify the instances from minority classes in imbalanced datasets. Common approaches to address this issue are to either assign higher misclassification costs to minority classes or replace the classification accuracy criterion with the area under the ROC curve which is a more relevant criterion when dealing with imbalanced datasets. Either way, the features should also be selected by an algorithm which is insensitive (robust) with respect to class distributions (otherwise the selected features may not be informative about minority classes, in the first place). Interestingly, by internally applying unequal class dependent costs, mutual information provides some robustness with respect to class distributions. Thus, even in an imbalanced case, a mutual information based feature selection algorithm is likely (though not guaranteed) to not overlook the features that represent the minority classes. In [Hu11], the concept of the mutual information classifier is investigated. Specifically, the internal cost matrix of the mutual information classifier is derived to show that it applies unequal misclassification costs when dealing with imbalanced data and showed that the mutual information classifier is an optimal classifier in the sense of maximizing a weighted classification accuracy rate. The following example shows this robustness.

Example 3.2. Assume an imbalanced binary classification task where $Pr(C{=}1) = 0.9$. As in Example 3.1, there are two binary features X_1 and X_2 and the goal is to select the optimum feature. Suppose $Pr(X_1{=}1|C{=}1) = 1$ and $Pr(X_1{=}-1|C{=}-1) = 0.5$. Unlike the first feature, X_2 can much better classify the minority class $Pr(X_2{=}-1|C{=}-1) = 1$ and $Pr(X_2{=}1|C{=}1) = 0.8$. It can be seen that the Bayes classifier with X_1 results in 100% classification rate for the majority class while only 50% correct classification for the minority. On the other hand, using X_2 leads to 100% correct classification for the minority class and 80% for the majority. Based on the probability of error, X_1 should be preferred since its probability of error is $P_{e1} = 0.05$ while $P_{e2} = 0.18$. However, by using X_1 the classifier cannot learn the rare event (50% classification rate) and thus randomly classifies the minority class

which is the class of interest in many applications. Interestingly, unlike the Bayesian error probabilities, mutual information prefers X_2 over X_1, since $I(X_2; C) = 0.20$ is greater than $I(X_1; C) = 0.18$. That is, mutual information is to some extent robust against imbalanced data.

Unfortunately, despite the theoretical appeal of the mutual information measure, given a limited amount of data, an accurate estimate of the mutual information would be impossible. Because to calculate mutual information, estimating the high-dimensional joint probability $Pr(\mathbf{X}, C)$ is inevitable which is, in turn, known to be an NP hard problem [KS01].

As mutual information is hard to evaluate, several alternatives have been suggested [Bat94], [PLD05], [KC02]. For instance, the Max-Relevance criterion approximates (3.1) with the sum of the mutual information values between individual features X_i and C:

$$\text{Max-Relevance} = \sum_{i=1}^{N} I(X_i; C) \tag{3.6}$$

Since it implicitly assumes that features are independent, it is likely that selected features are highly redundant. To overcome this problem, several heuristic corrective terms have been introduced to remove the redundant information and select mutually exclusive features. Here, it is shown that most of these heuristics are derived from the following expansions of mutual information with respect to X_i.

3.2.1 First Expansion: Multi-way Mutual Information

The first expansion of mutual information that is used here, relies on the natural extension of mutual information to more than two random variables proposed by McGill [McG54] and Abramson [Abr63]. According to their proposal, the three-way mutual information between random variables Y_i is defined by:

$$\begin{aligned} I(Y_1; Y_2; Y_3) =& I(Y_1; Y_3) + I(Y_2; Y_3) - I(Y_1, Y_2; Y_3) \\ =& I(Y_1; Y_2) - I(Y_1; Y_2|Y_3) \end{aligned} \tag{3.7}$$

where "," between variables denotes the joint variables. Note that, similar to two-way mutual information, it is symmetric with respect to Y_i variables, i.e.,

$I(Y_1;Y_2;Y_3) = I(Y_2;Y_3;Y_1)$. Generalizing over N variables:

$$\begin{aligned} I(Y_1;Y_2;\ldots;Y_N) = & I(Y_1;\ldots;Y_{N-1}) \\ & - I(Y_1;\ldots;Y_{N-1}|Y_N) \end{aligned} \tag{3.8}$$

Unlike 2-way mutual information, the generalized mutual information is not necessarily nonnegative and hence, can be interpreted as a signed measure of set intersection [Han80]. Consider (3.7) and assume Y_3 is class label C, then positive $I(Y_1;Y_2;C)$ implies that Y_1 and Y_2 are redundant with respect to C since $I(Y_1,Y_2;C) \leq I(Y_1;C) + I(Y_2;C)$. However, the more interesting case is when $I(Y_1;Y_2;C)$ is negative, i.e., $I(Y_1,Y_2;C) \geq I(Y_1;C) + I(Y_2;C)$. This means, the information contained in the interactions of the variables is greater than the sum of the information of the individual variables [Gur09].

$$\begin{aligned} I(\mathbf{X};C) = & \sum_{i_1=1}^{N} I(X_{i_1};C) - \sum_{i_1=1}^{N-1} \sum_{i_2=i_1+1}^{N} I(X_{i_1};X_{i_2};C) \\ & + \cdots + (-1)^{N-1} I(X_1;\ldots;X_N;C) \end{aligned} \tag{3.9}$$

An artificial example for this situation is the binary classification problem depicted in Figure 3.1, where the classification task is to discriminate between the ellipse class (class samples depicted by circles) and the line class (star samples) by using two features: values of x axis and values of y axis. As can be seen, since $I(x;C) \approx 0$ and $I(y;C) \approx 0$, there The mutual information in (3.1) can be expanded out in terms of generalized mutual information between the features and the class label as:

From the definition in (3.8) it is straightforward to infer this expansion. However, the more intuitive proof is to use the fact that mutual information is a measure of set intersection, i.e., $I(Y_1;Y_2;Y_3) = \mu(\mathbb{Y}_1 \cap \mathbb{Y}_2 \cap \mathbb{Y}_3)$, where

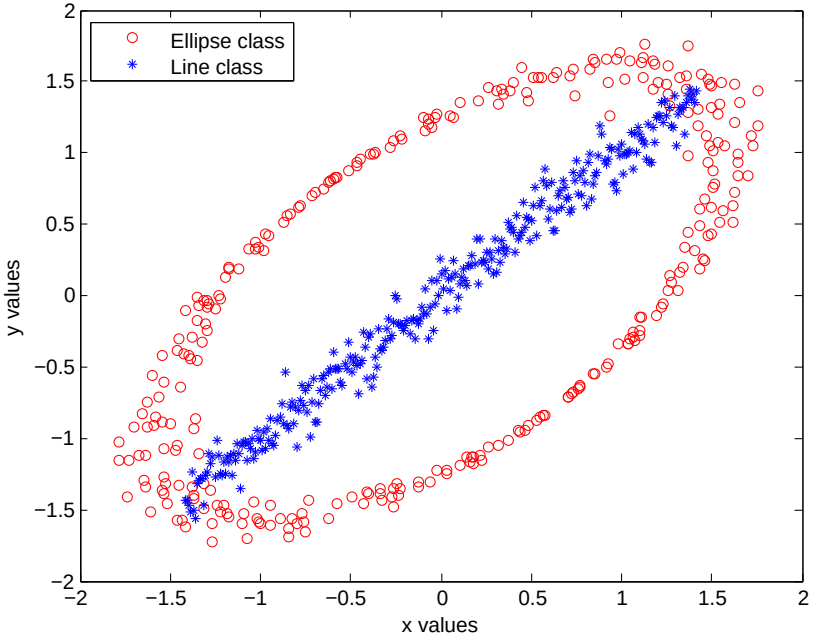

Figure 3.1: Synergy between x and y features. While information of each individual feature about the class label (ellipse or line) is almost zero, their joint information can almost completely remove the class label ambiguity.

$\mathbb{Y}_i$ is the corresponding event set of the Y_i variable. Now, expanding the N-variable measure function results in

$$
\begin{aligned}
I(\mathbf{X};C) &= \mu((\bigcup_{i=1}^{N}\mathbb{X}_i)\cap\mathbb{C}) = \mu(\bigcup_{i=1}^{N}(\mathbb{X}_i\cap\mathbb{C})) \qquad (3.10)\\
&= \sum_{i=1}^{N}\mu(\mathbb{X}_i\cap\mathbb{C}) - \sum_{i_1=1}^{N-1}\sum_{i_2=i_1+1}^{N}\mu(\mathbb{X}_{i_1}\cap\mathbb{X}_{i_2}\cap\mathbb{C})\\
&+\cdots+(-1)^{N-1}\mu(\mathbb{X}_1\cap\mathbb{X}_2\cdots\cap\mathbb{X}_N\cap\mathbb{C})
\end{aligned}
$$

where the last equation follows directly from the addition law or sum rule in set theory. The proof is complete by recalling that all measure functions with

the set intersection arguments in the last equation can be replaced by the mutual information functions according to the definition of mutual information in (3.2).

3.2.2 Second Expansion: Chain Rule of Information

The second expansion for mutual information is based on the *chain rule of information* [CT91]:

$$I(\mathbf{X};C) = \sum_{i=1}^{N} I(X_i;C|X_{i-1},\ldots,X_1) \tag{3.11}$$

The chain rule of information leaves the choice of ordering quite flexible. For example, the right side can be written in the order $(X_1, X_2, \ldots, X_N)$ or $(X_N, X_{N-1}, \ldots, X_1)$. In general, it can be expanded over $N!$ different permutations of the feature set $\{X_1, \ldots, X_N\}$. Taking the sum over all possible expansions yields,

$$\begin{aligned}(N!)I(\mathbf{X};C) &= (N-1)!\sum_{i=1}^{N} I(X_i;C) \\ &+ (N-2)!\sum_{i_1=1}^{N}\sum_{i_2\in\{1,\ldots,N\}\setminus i_1} I(X_{i_2};C|X_{i_1}) \\ &+ \cdots + (N-1)!\sum_{i=1}^{N} I(X_i;C|\{X_1,\ldots,X_N\}\setminus X_i)\end{aligned} \tag{3.12}$$

Dividing both sides by $(N-1)!/2$, and using the following equation $I(X_{i_1};C|X_{i_2}) = I(X_{i_1};C) - I(X_{i_1};X_{i_2};C)$ to replace $I(X_{i_1};C|X_{i_2})$ terms, our second expansion can be expressed as

$$\begin{aligned}\frac{N}{2}I(\mathbf{X};C) &= \sum_{i=1}^{N} I(X_i;C) \\ &- \frac{1}{N-1}\sum_{i_1=1}^{N-1}\sum_{i_2=i_1+1}^{N} I(X_{i_1};X_{i_2};C) \\ &+ \cdots + \frac{1}{2}\sum_{i=1}^{N} I(X_i;C|\{X_1,\ldots,X_N\}\setminus X_i)\end{aligned} \tag{3.13}$$

Ignoring the unimportant multiplicative constant $N/2$ on the left side of equation (3.13), the right side can be seen as a series expansion form of mutual information (up to a known constant factor).

3.2.3 Truncation of the Expansions

In the proposed expansions (3.9) and (3.13), mutual information terms with more than two features represent higher-order interaction properties. Neglecting the higher order terms yields the so-called truncated approximation of the mutual information function. If the constant coefficient in (3.13) is ignored, the truncated forms of suggested expansions can be written as:

$$D_1 = \sum_{i=1}^{N} I(X_i;C) - \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} I(X_i;X_j;C) \tag{3.14}$$

$$D_2 = \sum_{i=1}^{N} I(X_i;C) - \frac{1}{N-1} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} I(X_i;X_j;C) \tag{3.15}$$

where D_1 is the truncated approximation of (3.9) and D_2 is for (3.13). Interestingly, despite the very similar structure of the expressions in (3.14), they have intrinsically different behaviors. This difference seems to be rooted in different functional forms they employ to approximate the underlying high-order pdf with lower order distributions (i.e., how they combine these lower order terms). For instance, the functional form that MIFS employs to approximate $Pr(\mathbf{x})$ is shown in (3.19). While D_1 is not necessarily a positive value, D_2 is guaranteed to be a positive approximation since all terms in (3.12) are positive. However, D_2 may highly underestimate the mutual information values since it may violate the fact that (3.1) is always greater than or equal to $\max_i I(X_i;C)$.

JMI, mRMR & MIFS Criteria

Several known criteria including joint mutual information (JMI) [MB06], minimal Redundancy Maximal Relevance (mRMR) [PLD05] and Mutual Information Feature Selection (MIFS) [Bat94] can immediately be derived from D_1 and D_2.

Using the identity: $I(X_i;X_j;C) = I(X_i;C) + I(X_j;C) - I(X_i,X_j;C)$

in D_2 reveals that D_2 is equivalent to JMI.

$$\text{JMI} = D_2 = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} I(X_i, X_j; C) \tag{3.16}$$

Using $I(X_i; X_j; C) = I(X_i; X_j) - I(X_i; X_j|C)$ and ignoring the terms containing more than two variables, i.e., $I(X_i; X_j|C)$, in the second approximation D_2, one may immediately recognize the popular score function

$$\text{mRMR} = \sum_{i=1}^{N} I(X_i; C) - \frac{1}{N-1} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} I(X_i; X_j) \tag{3.17}$$

introduced by Peng et al. in [PLD05]. That is, mRMR is a truncated approximation of mutual information and not a heuristic approximation as suggested in [BPZL12].

The same line of reasoning as for mRMR can be applied to D_1 to achieve MIFS with β equal to 1.

$$\text{MIFS} = \sum_{i=1}^{N} I(X_i; C) - \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} I(X_i; X_j) \tag{3.18}$$

Observation: A constant feature is a potential danger for the above measures. While adding an informative but correlated feature may reduce the score value (since $I(X_i; X_j|C) - I(X_i; X_j)$ can be negative), adding a non-informative constant feature Z to a feature set does not reduce its score value since both $I(Z; C)$ and $I(Z; X_i; C)$ terms are zero, that is, constant features may be preferred over informative but correlated features. Therefore, it is essential to remove constant features by some pre-processing before using the above criteria for feature selection.

Implicitly Assumed Distribution

An obvious question arising in this context with respect to the proposed truncated approximations is: Under what probabilistic assumptions do the proposed approximations become valid mutual information functions? That is, which structure should a joint pdf admit, to yield mutual information in the forms of D_1 or D_2?

For instance, if we assume features are mutually and class conditionally independent, i.e., $Pr(\mathbf{X}) = \prod_{i=1}^{N} Pr(X_i)$ and $Pr(\mathbf{X}, C) = Pr(C)\prod_{i=1}^{N} Pr(X_i|C)$, then it is easy to verify that mutual information has the form of Max-Relevance introduced in (3.6). These two assumptions, define the adapted *independence-map* of $Pr(\mathbf{X}, C)$ where the independence-map of a joint probability distribution is defined as follows.

Definition 3.3. *An independence-map (i-map) is a look up table or a set of rules that denote all the conditional and unconditional independence between random variables. Moreover, an i-map is consistent if it leads to a valid factorized probability distribution.*

That is, given a consistent i-map, a high-order joint probability distribution is approximated with product of low-order pdfs and the obtained approximation is a valid pdf itself (e.g., $\prod_{i=1}^{N} Pr(X_i)$ is an approximation of the high-order pdf $Pr(\mathbf{X})$ and it is also a valid probability distribution).

The question regarding the implicit consistent i-map that MIFS adopts has been investigated in [BP10]. However, the assumption set (i-map) suggested in their work is inconsistent and leads to the incorrect conclusion that MIFS upper bounds the Bayesian classification error via the inequality (3.4). As shown in the following theorem, unlike the Max-Relevance case, there is no i-map that can produce mutual information in the forms of mRMR or MIFS (ignoring the trivial solution that reduces mRMR or MIFS to Max-Relevance).

Theorem 3.4. *There is no consistent i-map other than the trivial solution, i.e., the i-map indicating that random variables are mutually and class conditionally independent, that can produce mutual information functions in the forms of mRMR (3.17) or MIFS (3.18) for an arbitrary number of features.*

Proof: The proof is by contradiction. Suppose there is a consistent i-map, where its corresponding joint pdf $\hat{Pr}(\mathbf{X}, C)$ (which is the approximation of $Pr(\mathbf{X}, C)$) can generate mutual information in the forms of (3.17) or (3.18). That is, if this i-map is adopted, by replacing $\hat{Pr}(\mathbf{X}, C)$ in (3.1) we get mRMR or MIFS. This implies that mRMR and MIFS are *always* valid set measures for all datasets regardless of their true underlying joint probability distributions. Now, if it is shown (by any example) that they are not valid mutual information measures, i.e., they are not always positive and monotonic, then the assumption that $\hat{Pr}(\mathbf{X}, C)$ exists and is a valid pdf has been contradicted. It is not so difficult to construct an example in which mRMR or MIFS

can get negative values. Consider the case where features are independent of class label, $I(X_i; C) = 0$, while they have nonzero dependencies among themselves, $I(X_i; X_j) \neq 0$. In this case, both mRMR and MIFS generate negative values which is not allowed by a valid set measure. This contradicts our assumption that they are generated by a valid distribution, so we are forced to conclude that there is no consistent i-map that results in mutual information in the mRMR or MIFS forms.■

The same line of reasoning can be used to show that D_1 and D_2 are also not valid measures.

However, despite the fact that no valid pdf can produce mutual information of those forms, it is still valid to ask for which low-order approximations of the underlying high-order pdfs, mutual information reduces to a truncated approximation form. That is, we do not restrict an approximation to be a valid distribution anymore. Any functional form of low-order pdfs may be seen as an approximation of the high-order pdfs and may give rise to MIFS or mRMR. In the next following, these assumptions for the MIFS criterion are revealed.

MIFS Derivation from Kirkwood Approximation

It is shown in [KKG07] that truncation of the joint entropy $H(\mathbf{X})$ at the rth-order is equivalent to approximating the full-dimensional pdf $Pr(\mathbf{X})$ using joint pdfs with dimensionality of r or smaller. This approximation is called rth order Kirkwood approximation. The truncation order that is chosen, partially determines our belief about the structure of the function that approximates the $Pr(\mathbf{X})$.

The 2nd order Kirkwood approximation of $Pr(\mathbf{X})$, can be denoted as follows [KKG07]:

$$\hat{Pr}(\mathbf{X}) = \frac{\prod_{i=1}^{N-1} \prod_{j=i+1}^{N} Pr(X_i, X_j)}{\left[\prod_{i=1}^{N} Pr(X_i)\right]^{N-2}} \tag{3.19}$$

Now, assume the following two assumptions hold:

Assumption 3.5. *Features are class conditionally independent, that is:* $Pr(\mathbf{X}|C) = \prod_{i=1}^{N} Pr(X_i|C)$

Assumption 3.6. $Pr(\mathbf{X})$ *is well approximated by a 2nd order Kirkwood superposition approximation in (3.19).*

Then, writing the definition of mutual information and applying the above assumptions yields the MIFS criterion

$$
\begin{aligned}
I(\mathbf{X};C) &= H(\mathbf{X}) - H(\mathbf{X}|C) \\
&\overset{(a)}{\approx} \sum_{i=1}^{N} H(X_i) - \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} I(X_i;X_j) - H(\mathbf{X}|C) \\
&\overset{(b)}{=} \sum_{i=1}^{N} I(X_i;C) - \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} I(X_i;X_j)
\end{aligned}
\tag{3.20}
$$

In the above equation, (a) follows the second assumption by substituting the 2nd order Kirkwood approximation (3.19) inside the logarithm of the entropy integral and (b) is an immediate consequence of the first assumption.

The first assumption has already appeared in previous works [BPZL12] [BP10]. However, the second assumption is novel and, to the best of our knowledge, the connection between the Kirkwood approximation and the MIFS criterion has not been explored before.

It is worth to mention that, in reality, both assumptions can be violated. Specifically, the Kirkwood approximation may not precisely reproduce dependencies which might be observed in real-world datasets. Moreover, it is important to remember that the Kirkwood approximation is not a valid probability distribution.

3.2.4 The superiority of the D_2 Approximation

Measure D_2 always yields a positive score and generally tends to underestimate the mutual information while D_1 shows a large overestimation for independent features and a large underestimation (even becoming negative) in the presence of dependent features. In general, D_2 shows more robustness than D_1. The same results can be observed for mRMR which is derived from D_2 and MIFS derived from D_1. Previous works also arrived to the same results and reported that mRMR performs better and more robustly than MIFS especially when the feature set is large. Therefore, in the following sections D_2 is used as the truncated approximation. For simplicity, its subscript is dropped and it is rewritten as follows:

$$D(\{X_1,\ldots,X_N\}) = \sum_{i=1}^{N} I(X_i;C) - \frac{1}{N-1}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} I(X_i;X_j;C) \tag{3.21}$$

Note that although D in (3.21) is not a formal set measure any more, it still can be seen as a score function for sets. However, it is noteworthy that unlike formal measures, the suggested approximations are no longer monotonic where the monotonicity merely means that a subset of features should not be better than any larger set that contains the very same subset. Therefore, as explained in [NF77] the branch and bound based search strategies cannot be applied to them.

A very similar approach has been applied [Bro09] (by using D_1 approximation) to derive several known criteria like MIFS [Bat94] and mRMR [PLD05]. However, in [Bro09] and most of other previous works, the set score function in (3.21) is immediately reduced to an individual-feature score function by fixing $N-1$ features in the feature set. This will let them to run a greedy selection search method over the feature set which essentially is a one-feature-at-a-time selection strategy. It is clearly a naive approximation of the optimal NP-hard search algorithm and may perform poorly under some conditions. In the following, a convex approximation of the binary objective function appearing in feature selection is investigated. This approximation is inspired by the Goemans-Williamson maximum cut approximation approach [GW95].

3.3 Search Strategies

Given a measure function[1] D, the subset selection problem (SSP) can be defined as follows:

Definition 3.7. *Given N features X_i and a dependent variable C, select a subset of $P \ll N$ features that maximizes the measure function. Here it is assumed that the cardinality P of the optimal feature subset is known.*

[1] By some abuse of terminology, any set function in this section is referred to as a measure, no matter whether they satisfy the formal measure properties.

In practice, the exact value of P can be obtained by evaluating subsets for different values of cardinality P with the final induction algorithm. Note that it is intrinsically different than wrapper methods. While in wrapper methods 2^N subsets have to be tested, here at most N runs of the learning algorithm are required to evaluate all possible values of P.

A search strategy is an algorithm trying to find a feature subset in the feature subset space with 2^N members[2] that optimizes the measure function. The wide range of proposed search strategies in the literature can be divided into three categories:

- Exponential complexity methods including exhaustive search [Koh96], branch and bound based algorithms [NF77].
- Sequential selection strategies with two very popular members, forward selection and backward elimination methods.
- Stochastic methods like simulated annealing and genetic algorithms [VD93], [Doa92].

Here, a fourth class of search strategies is introduced which is based on the convex relaxation of the 0-1 integer programming and explore its approximation ratio by establishing a link between SSP and an instance of the maximum-cut problem in graph theory. In the following, the two popular sequential search methods are briefly discussed and the proposed solution is represented: a close to optimal polynomial-time complexity search algorithm and its evaluation on different datasets.

3.3.1 Convex Based Search

The forward selection (FS) algorithm selects a set $\mathbb{S}$ of size P iteratively as follows:

1. Initialize $\mathbb{S}_0 = \emptyset$.
2. In each iteration i, select the feature X_m maximizing $D(\mathbb{S}_{i-1} \cup X_m)$, and set $\mathbb{S}_i = \mathbb{S}_{i-1} \cup X_m$.
3. Output $\mathbb{S}_P$.

[2] Given a P, the size of the feature subset space reduces to $\binom{N}{P}$.

Similarly, backward elimination (BE) can be described as:

1. Start with the full set of features $\mathbb{S}_N$.
2. Iteratively remove a feature X_m maximizing $D(\mathbb{S}_i \backslash X_m)$, and set $\mathbb{S}_{i-1} = \mathbb{S}_i \backslash X_m$, where removing X from $\mathbb{S}$ is denoted by $\mathbb{S} \backslash X$.
3. Output $\mathbb{S}_P$.

An experimentally comparative evaluation of several variants of these two algorithms has been conducted in [AB96]. From an information theoretical standpoint, the main disadvantage of the forward selection method is that it only can evaluate the utility of a single feature in the limited context of the previously selected features. The artificial binary classifier in Figure 3.1 may illustrate this issue. Since the information content of each feature (x and y) is almost zero, it is highly probable that the forward selection method fails to select them in the presence of some other more informative features.

Contrary to forward selection, backward elimination can evaluate the contribution of a given feature in the context of all other features. Perhaps this is why it has been frequently reported to show superior performance than forward selection. However, its overemphasis on feature interactions is a double-edged sword and may lead to a sub-optimal solution.

Example 3.8. Imagine a four dimensional feature selection problem where X_1 and X_2 are class conditionally and mutually independent of X_3 and X_4, i.e., $Pr(X_1, X_2, X_3, X_4) = Pr(X_1, X_2)Pr(X_3, X_4)$ and $Pr(X_1, X_2, X_3, X_4|C) = Pr(X_1, X_2|C)Pr(X_3, X_4|C)$. Consider $I(X_1; C)$ and $I(X_2; C)$ are equal to zero, while their interaction is informative. That is, $I(X_1, X_2; C) = 0.4$. Moreover, assume $I(X_3; C) = 0.2$, $I(X_4; C) = 0.25$ and $I(X_3, X_4; C) = 0.45$. The goal is to select only two features out of four. Here, backward elimination will select $\{X_1, X_2\}$ rather than the optimal subset $\{X_3, X_4\}$ because, removing either of X_1 or X_2 will result in 0.4 reduction of the mutual information value $I(X_1, \ldots, X_4; C)$, while eliminating X_3 or X_4 deducts at most 0.25.

One may draw the conclusion that backward elimination tends to sacrifice the individually-informative features in favor of the merely cooperatively-informative features. As a remedy, several hybrid forward-backward sequential search methods have been proposed. However, they all fail in one way or another and more importantly cannot guarantee the goodness of the solution.

Alternatively, a sequential search method can be seen as an approximation of the combinatorial subset selection problem. To propose a new approximation method, the underlying combinatorial problem has to be studied. To this end, the SSP defined in the beginning of this section is reformulated as:

$$\begin{aligned} &\max_{\mathbf{x}} \ \mathbf{x}^T \mathbf{Q} \mathbf{x} \\ &\sum_{i=1}^{N} x_i = P \\ &x_i \in \{0,1\} \text{ for } i = 1, \ldots, N \end{aligned} \tag{3.22}$$

where $\mathbf{Q}$ is a symmetric mutual information matrix constructed from the mutual information terms in (3.21):

$$\mathbf{Q} = \begin{pmatrix} I(X_1;C) & \cdots & -\frac{\lambda}{2} I(X_1;X_N;C) \\ -\frac{\lambda}{2} I(X_1;X_2;C) & \cdots & -\frac{\lambda}{2} I(X_2;X_N;C) \\ \vdots & \ddots & \vdots \\ -\frac{\lambda}{2} I(X_1;X_N;C) & \cdots & I(X_N;C) \end{pmatrix} \tag{3.23}$$

where $\lambda = \frac{1}{P-1}$ and $\mathbf{x} = [x_1, \ldots, x_N]$ is a binary vector where the variables x_i are set-membership binary variables indicating the presence of the corresponding features X_i in the feature subset. It is straightforward to verify that for any binary vector $\mathbf{x}$, the objective function in (3.22) is equal to the score function $D(\mathbb{X}_{nz})$ where $\mathbb{X}_{nz} = \{X_i | x_i = 1; i = 1, \ldots, N\}$. Note that, for mRMR $I(X_i; X_j; C)$ terms have to be replaced with $I(X_i; X_j)$.

The (0,1)-quadratic programming problem (3.22) has attracted a great deal of theoretical study because of its importance in combinatorial problems (see [PRW95] and references therein). This problem can simply be transformed to a (-1,1)-quadratic programming problem,

$$\begin{aligned} &\max_{\mathbf{y}} \ \frac{1}{4} \mathbf{y}^T \mathbf{Q} \mathbf{y} + \frac{1}{2} \mathbf{y}^T \mathbf{Q} \mathbf{e} + c \\ &\sum_{i=1}^{N} y_i = 2P - N \\ &y_i \in \{-1,1\} \text{ for } i = 1, \ldots, N \end{aligned} \tag{3.24}$$

via the transformation $\mathbf{y} = 2\mathbf{x} - \mathbf{e}$, where $\mathbf{e}$ is an all ones vector. Additionally c in the above formulation is a constant equal to $\frac{1}{4} \mathbf{e}^T \mathbf{Q} \mathbf{e}$ and it can be ignored

because it is independent of $\mathbf{y}$. In order to homogenize the objective function in (3.24), an $(N+1)\times(N+1)$ matrix $\mathbf{Q}^u$ is defined by adding a 0-th row and column to $\mathbf{Q}$ so that:

$$\mathbf{Q}^u = \begin{pmatrix} 0 & \mathbf{e}^T\mathbf{Q} \\ \mathbf{Q}^T\mathbf{e} & \mathbf{Q} \end{pmatrix} \tag{3.25}$$

Ignoring the constant factor $\frac{1}{4}$ in (3.24), the equivalent homogeneous form of (3.22) can be written as:

$$\langle \text{SSP} \rangle \qquad \begin{aligned} & S_{\text{SSP}} = \max_{\mathbf{y}} \mathbf{y}^T\mathbf{Q}^u\mathbf{y} \\ & \sum_{i=1}^{N} y_i y_0 = 2P - N \\ & y_i \in \{-1, 1\} \text{ for } i = 0, \ldots, N \end{aligned} \tag{3.26}$$

Note that $\mathbf{y}$ is now an $N+1$ dimensional vector with the first element $y_0 = \pm 1$ as a reference variable. Given the solution of the problem above, i.e., $\mathbf{y}$, the optimal feature subset is obtained by $\mathbb{X}_{op} = \{X_i | y_i = y_0\}$.

The optimization problem in (3.26) can be seen as an instance of the maximum-cut problem [GW95] with an additional cardinality constraint, also known as the k-heaviest subgraph or maximum partitioning graph problem. The two main approaches to solve this combinatorial problem are either to use the linear programming relaxation by linearizing the product of two binary variables [FY83], or the semi-definite programming (SDP) relaxation suggested in [GW95]. The SDP relaxation has been proved to have exceptionally high performance and achieves the approximation ratio of 0.878 for the original maximum-cut problem. The SDP relaxation of (3.26) is:

$$\langle \text{SDP} \rangle \qquad \begin{aligned} & S_{\text{SDP}} = \max_{\mathbf{Y}} \text{tr}\{\mathbf{Q}^u\mathbf{Y}\} \\ & \sum_{i,j=1}^{N} Y_{ij} = (2P - N)^2 \\ & \sum_{i=1}^{N} Y_{i0} = (2P - N) \\ & \text{diag}(\mathbf{Y}) = \mathbf{e} \\ & \mathbf{Y} \succeq 0 \end{aligned} \tag{3.27}$$

where $\mathbf{Y}$ is an unknown $(N+1) \times (N+1)$ positive semi-definite matrix and tr$\{\mathbf{Y}\}$ denotes its trace. Obviously, any feasible solution $\mathbf{y}$ for $\langle$SSP$\rangle$is also feasible for its SDP relaxation by $\mathbf{Y} = \mathbf{y}\mathbf{y}^T$. Furthermore, it is not difficult to see that any rank one solution, rank$(\mathbf{Y}) = 1$, of $\langle$SDP$\rangle$is a solution of $\langle$SSP$\rangle$. The $\langle$SDP$\rangle$ problem can be solved within an additive error γ of the optimum by for example interior point methods [BV04] whose computational complexity are polynomial in the size of the input and $log(\frac{1}{\gamma})$. However, since its solution is not necessarily a rank one matrix, some more steps are required to obtain a feasible solution for $\langle$SSP$\rangle$. Algorithm 3.1 summarizes the approximation algorithm for $\langle$SSP$\rangle$ which in the following will be referred to as convex based relaxation approximation (COBRA) algorithm.

Algorithm 3.1: COBRA Feature Selection

Input: number of repetitions T and SDP parameters in 3.27.
For $t = 1, \ldots, T$ **do**

(a) **Randomized rounding**: Using the multivariate normal distribution with a zero mean and a covariance matrix $\mathbf{R} = \mathbf{Y}_{SDP}$ to sample $\mathbf{u}$ from distribution $\mathcal{N}(0, \mathbf{R})$ and construct $\hat{\mathbf{x}} = \text{sign}(\mathbf{u})$. Select $\mathbb{X}_t = \{X_i | \hat{x}_i = \hat{x}_0\}$

(b) **Size adjustment**: By using the greedy forward or backward algorithm, resize the cardinality of $\mathbb{X}_t$ to P.

End
Output: Ouput $\mathbb{X}_t$ with the maximum SDP score.

The randomized rounding step in COBRA is a standard procedure to produce a binary solution from the real-valued solution of $\langle$SDP$\rangle$ and is widely used for designing and analyzing approximation algorithms [Rag88]. The third step is to construct a feasible solution that satisfies the cardinality constraint. Generally, it can be skipped since in feature selection problems the exact satisfaction of the cardinality constraint is not required.

The SDP-NAL solver [ZST10] was used with the Yalmip interface [Lof04] to implement this algorithm in Matlab. SDP-NAL uses the Newton augmented Lagrangian method to efficiently solve SDP problems. It can solve large scale problems (N up to a few thousand) in an hour on a PC with an Intel Core i7 CPU. Even more efficient algorithms for low-rank SDP have been suggested claiming that they can solve problems with the size up to N=30000 in a reasonable amount of time [GPP$^+$12]. Here only the SDP-NAL solver was used in the experiments.

3.3.2 Approximation Analysis

In order to gain more insight into the quality of a measure function, it is essential to be able to directly examine it. However, since estimating the exact mutual information value in real data is not feasible, it is not possible to directly evaluate the measure function. Its quality can only be indirectly examined through the final classification performance (or other measurable criteria).

However, the quality of a measure function is not the only contributor to the classification rate. Since SSP is an NP-hard problem, the search strategy can only find a local optimal solution. That is, besides the quality of a measure function, the inaccuracy of the search strategy also contributes to the final classification error. Thus, in order to draw a conclusion concerning the quality of a measure function, it is essential to have an insight about the accuracy of the search strategy in use. In this section, the accuracy of the proposed method with the traditional backward elimination approach is compared.

A standard approach to investigate the accuracy of an optimization algorithm is by analyzing how close it gets to the optimal solution. Unfortunately, feature selection is an NP-hard problem and thus achieving the optimal solution to use as reference is only feasible for small-sized problems. In such cases, one wants a provable solution's quality and certain properties about the algorithm, such as its approximation ratio. Given a maximization problem, an algorithm is called ρ-approximation algorithm if the approximate solution is at least ρ times the optimal value. That is, in our case $\rho S_{SSP} \leq S_{COBRA}$, where $S_{COBRA} = D(\mathbb{X}_{COBRA})$. The factor ρ is usually referred to as the approximation ratio in the literature.

The approximation ratios of BE and COBRA can be found by linking the SSP to the k-heaviest subgraph problem (k-HSP) in graph theory. k-HSP is an instance of the max-cut problem with a cardinality constraint on the selected subset, that is, to determine a subset S of k vertices such that the weight of the subgraph induced by S is maximized [SW98]. From the definition of k-HSP, it is clear that SSP with the criterion (3.21) is equivalent to the P-heaviest subgraph problem since it selects the heaviest subset of features with the cardinality P, where heaviness of a set is the score assigned to it by D.

An SDP based algorithm for k-HSP has been suggested in [SW98] and its approximation ratio has been analyzed. Their results are directly applicable to COBRA since both algorithms use the same randomization method (step 2 of COBRA) and the randomization is the main ingredient of their approximation analysis. The approximation ratio of BE for k-HSP has been investigated in

Values of P	$N/2$	$N/3$	$N/4$	$N/6$	$N/8$	$N/10$	$N/20$
BE	0.4	0.25	0.16	0.10	0.071	0.055	0.026
COBRA	0.48	0.33	0.24	0.13	0.082	0.056	0.015

Table 3.1: Approximation ratios of backward elimination (BE) and COBRA for different N/P values.

[AITT00]. It is a deterministic analysis and their results are also valid for our case, i.e., using BE for maximizing D.

The approximation ratios of both algorithms for different values of P, as a function of N (total number of features), have been listed in Table 3.1 (values are calculated from the formulas in [AITT00]). As can be seen, as P becomes smaller, the approximation ratio approaches zero yielding the trivial lower bound 0 on the approximate solution. However, for larger values of P, the approximation ratio is nontrivial since it is bounded away from zero. For all cases shown in the table except the last one, COBRA gives better guarantee bound than BE. Thus, we may conclude that COBRA is more likely to achieve better approximate solution than BE.

In the focus of the following section is on comparing the proposed search algorithm with sequential search methods in conjunction with different measure functions and over different classifiers and datasets.

3.4 Experiments for COBRA Evaluation

The evaluation of a feature selection algorithm is an intrinsically difficult task since there is no direct way to evaluate the goodness of a *selection process* in general. Thus, usually a selection algorithm is scored based on the performance of its output, i.e., the selected feature subset, in some specific classification (regression) system. This kind of evaluation can be referred to as the goal-dependent evaluation. However, this method obviously cannot evaluate the generalization power of the selection process on different induction algorithms. To evaluate the generalization strength of a feature selection algorithm, a goal-independent evaluation is required. Thus, for evaluation of the feature selection algorithms, it is proposed to compare the algorithms over different datasets with multiple classifiers. This method leads to a more

Dataset Name	Mnemonic	# Features	# Samples	# Classes
Arrhythmia	ARR	278	370	2
NCI	NCI	9703	60	9
DBWorld e-mails	DBW	4702	64	2
CNAE-9	CNA	856	1080	9
Internet Adv.	IAD	1558	3279	2
Madelon	MAD	500	2000	2
Lung Cancer	LNG	56	32	3
Dexter	DEX	20000	300	2

Table 3.2: Dataset descriptions

classifier-independent evaluation process.

Algorithm 3.3: Estimating P by searching over an admissible set that minimizes the classification error-rate.

Input: P: $\mathbb{P}=\{P_1,\ldots,P_L\}$.

For P in $\mathbb{P}$ **do**

(a) Run the COBRA algorithm and output the solution $\mathbb{X}$.

(b) Derive the classifier error-rate by applying K-fold.

(c) Cross-validation and save the classification accuracy $CL(P)$.

End

Output: $P_{opt} = \underset{P}{\operatorname{argmin}}\, CL(P)$

Some properties of the eight datasets used in the experiments are listed in Table 3.2. All datasets are available on the UCI machine learning archive [FA10], except the NCI data which can be found in the website of Peng et al. [PLD05]. These datasets have been widely used in previous feature selection studies [PLD05], [CSO10]. The goodness of each feature set was evaluated with five classifiers including support vector machine (SVM), random forest (RF), classification and regression tree (CART), neural network (NN) and linear discriminant analysis (LDA). To derive the classification accuracies, 10-fold cross-validation is performed except for the NCI, DBW and LNG datasets where leave-one-out cross-validation is used.

Datasets	**NCI**	**DBW**	**IAD**	**CNA**
S-ratio	0.814±0.031	0.898±0.024	0.889±0.027	0.972±0.010
Datasets	**MAD**	**LNG**	**ARR**	**DEX**
S-ratio	0.892±0.0283	0.713±0.033	0.677±0.056	0.951±0.019

Table 3.3: Mean and standard deviation of similarity ratio of feature subsets from the COBRA algorithm. For each dataset (except LNG) 100 similarity ratios were evaluated for $P = 10, ..., 109$. For the LNG dataset, which only contains 56 features, the average was taken over $P = 5, ..., 49$.

As explained before, filter-based methods consist of two components: A measure function and a search strategy. The measure functions utilized in the experiments are mRMR and JMI defined in (3.17) and (3.16), respectively. To unambiguously refer to an algorithm, it is denoted by measure function + search method used in that algorithm, eg., mRMR+FS.

A simple algorithm listed in Algorithm 3.3 is employed to search for the optimal value of the subset cardinality P, where P ranges over a set $\mathbb{P}$ of admissible values. In the worst case, $\mathbb{P} = \{1, \ldots, N\}$.

Table 3.4 and Table 3.5 show the results obtained for the 8 datasets and 5 classifiers. Friedman test with the corresponding Wilcoxon-Nemenyi post-hoc analysis was used to compare the different algorithms. However, looking at the classification rates even before running the Friedman tests on them reveals a few interesting points which are marked in bold font.

First, on the small size datasets (NCI, DBW and LNG), mRMR+COBRA consistently shows higher performance than other algorithms. The reason lies in the fact that the *similarity ratio* of the feature sets selected by COBRA is lower than BE or FS feature sets. The *similarity ratio* of two consecutive sets $\mathbb{X}_i$ and $\mathbb{X}_j$, with $j = i + 1$ is defined as

$$S_i = \frac{|\mathbb{X}_i \cap \mathbb{X}_j|}{|\mathbb{X}_i|} \tag{3.28}$$

Table 3.3 reports the average of the similarity ratios of 100 subsequent feature sets ($\frac{1}{100}\sum_{i=10}^{109} S_i$) for the datasets. From the definition of similarity ratio it is clear that for BE and FS this ratio is always equal to 1. However, because of the randomization step this ratio may widely vary for COBRA. That is, COBRA generates quite diverse feature sets. Some of these feature sets have relatively low scores as compared with BE or FS sets. However, since for

Classifiers	SVM	LDA	CART	RF	NN	Average
			NCI Dataset			
mRMR+COBRA	(54) 81.7	(95) 78.3	(20) 45.0	(71) 88.3	(60) 75.0	**73.67**
mRMR+FS	(32) 78.3	(11) 68.3	(2) 45.0	(12) 83.3	(99) 70.0	69.00
mRMR+BE	(26) 76.6	(11) 68.3	(2) 45.0	(13) 85.0	(31) 71.7	69.33
JMI+COBRA	(72) 85.0	(70) 75.0	(28) 45.0	(45) 90.0	(93) 75.0	74.00
JMI+FS	(27) 75.0	(17) 68.3	(82) 45.0	(17) 86.6	(78) 70.0	69.00
JMI+BE	(23) 76.6	(20) 76.6	(7) 33.3	(19) 86.6	(89) 76.6	70.00
			DBW Dataset			
mRMR+COBRA	(38) 96.9	(152)92.2	(38) 86.0	(33) 92.2	(33) 98.4	**93.12**
mRMR+FS	(31) 93.7	(4) 89.0	(4) 86.0	(7) 90.6	(9) 92.2	90.31
mRMR+BE	(110)93.7	(6) 89.0	(4) 82.8	(29) 92.2	(9) 92.2	90.00
JMI+COBRA	(35) 93.7	(14) 89.0	(8) 82.8	(24) 92.2	(108)93.7	90.31
JMI+FS	(23) 93.7	(6) 89.0	(5) 82.8	(34) 92.2	(96) 92.2	90.00
JMI+BE	(24) 93.7	(6) 89.0	(5) 82.8	(23) 92.2	(149)92.2	90.00
			CNA Dataset			
mRMR+COBRA	(200)94.0	(183)92.7	(63) 75.0	(183)90.8	(187)92.0	88.91
mRMR+FS	(149)90.6	(142)90.4	(7) 70.2	(138)87.7	(78) 85.5	84.88
mRMR+BE	(199)94.0	(165)92.5	(47) 75.0	(176)90.8	(84) 92.2	88.90
JMI+COBRA	(140)92.6	(146)92.2	(47) 75.0	(148)90.4	(148)91.4	88.30
JMI+FS	(150)92.7	(142)92.1	(48) 75.3	(148)90.7	(145)91.3	88.40
JMI+BE	(150)92.7	(142)92.1	(48) 75.0	(144)90.4	(134)91.2	88.30
			IAD Dataset			
mRMR+COBRA	(165)96.5	(140)96.1	(28) 96.4	(160)97.2	(68) 97.1	96.64
mRMR+FS	(109)96.2	(127)95.8	(127)96.7	(25) 97.0	(52) 97.2	96.58
mRMR+BE	(22) 96.3	(163)95.9	(121)96.1	(109)97.2	(148)97.4	96.58
JMI+COBRA	(112)96.3	(4) 96.3	(9) 96.3	(57) 97.3	(140) 100	97.24
JMI+FS	(9) 96.2	(4) 96.2	(52) 96.4	(7) 96.8	(7) 97.8	96.68
JMI+BE	(4) 96.6	(17) 95.8	(79) 96.3	(13) 96.5	(10) 97.2	96.48

Table 3.4: Comparison of COBRA with the greedy search methods over different datasets. For each classifier and combination of search method and measure function, the values in parentheses is the number of selected features and the second value is the classification accuracy. The last column reports the average of the classification accuracies for each algorithm. Bold numbers represent statistically significant test results where COBRA outperforms sequential feature selection methods.

Classifiers	SVM	LDA	CART	RF	NN	Average
			MAD Dataset			
mRMR+COBRA	(12) 83.2	(13) 60.4	(26)80.5	(12) 88.0	(11) 62.2	**74.81**
mRMR+FS	(32) 55.3	(5) 55.5	(12)58.2	(49) 57.3	(5) 52.7	55.82
mRMR+BE	(14) 55.3	(11) 54.8	(31)57.3	(26) 56.4	(115)48.6	54.50
JMI+COBRA	(13) 82.5	(12) 60.7	(40)80.7	(13) 87.6	(4) 61.1	74.54
JMI+FS	(13) 82.5	(12) 60.7	(58)80.5	(13) 87.9	(19) 59.2	74.20
JMI+BE	(13) 82.5	(12) 60.7	(58)80.5	(13) 87.3	(20) 60.1	74.25
			LNG Dataset			
mRMR+COBRA	(23) 75.0	(28) 96.9	(13)71.8	(28) 68.7	(27) 71.8	**76.87**
mRMR+FS	(7) 81.2	(5) 68.7	(5) 71.8	(5) 75.0	(6) 71.8	73.75
mRMR+BE	(7) 81.2	(4) 68.7	(4) 71.8	(4) 75.0	(4) 75.0	74.37
JMI+COBRA	(7) 78.1	(6) 71.8	(5) 71.8	(5) 75.0	(5) 68.7	73.12
JMI+FS	(7) 78.1	(4) 71.8	(4) 71.8	(8) 78.1	(5) 68.7	73.75
JMI+BE	(7) 78.1	(6) 71.8	(5) 71.8	(6) 78.1	(6) 71.8	74.37
			ARR Dataset			
mRMR+COBRA	(45) 81.9	(48) 76.3	(30)75.4	(43) 82.2	(57) 72.9	77.75
mRMR+FS	(34) 81.3	(43) 76.1	(7) 78.3	(34) 81.3	(5) 75.7	78.56
mRMR+BE	(36) 81.6	(43) 76.3	(22)78.0	(25) 82.9	(8) 76.1	79.02
JMI+COBRA	(26) 80.6	(51) 74.7	(15)78.3	(51) 81.5	(13) 71.9	77.41
JMI+FS	(47) 74.3	(38) 73.5	(26)76.9	(37) 79.2	(54) 70.0	74.80
JMI+BE	(47) 74.3	(38) 73.5	(26)76.9	(25) 80.0	(29) 68.6	74.66
			DEX Dataset			
mRMR+COBRA	(3) 92.0	(131)86.3	(24)80.7	(3) 93.0	(3) 81.3	86.66
mRMR+FS	(3) 90.3	(56) 87.0	(94)80.3	(3) 92.0	(3) 80.0	86.00
mRMR+BE	(3) 90.0	(131)87.3	(18)80.3	(3) 91.6	(99) 78.6	85.53
JMI+COBRA	(88) 91.6	(13) 83.0	(12)80.3	(3) 94.0	(3) 81.0	86.00
JMI+FS	(149)91.0	(129)87.6	(95)80.3	(119)92.3	(94) 80.6	86.40
JMI+BE	(149)90.0	(128)87.3	(22)81.0	(146)92.0	(138)78.0	85.60

Table 3.5: Comparison of COBRA with the greedy search methods over different datasets. For each classifier and combination of search method and measure function, the values in parentheses is the number of selected features and the second value is the classification accuracy. The last column reports the average of the classification accuracies for each algorithm. Bold numbers represent statistically significant test results where COBRA outperforms sequential feature selection methods.

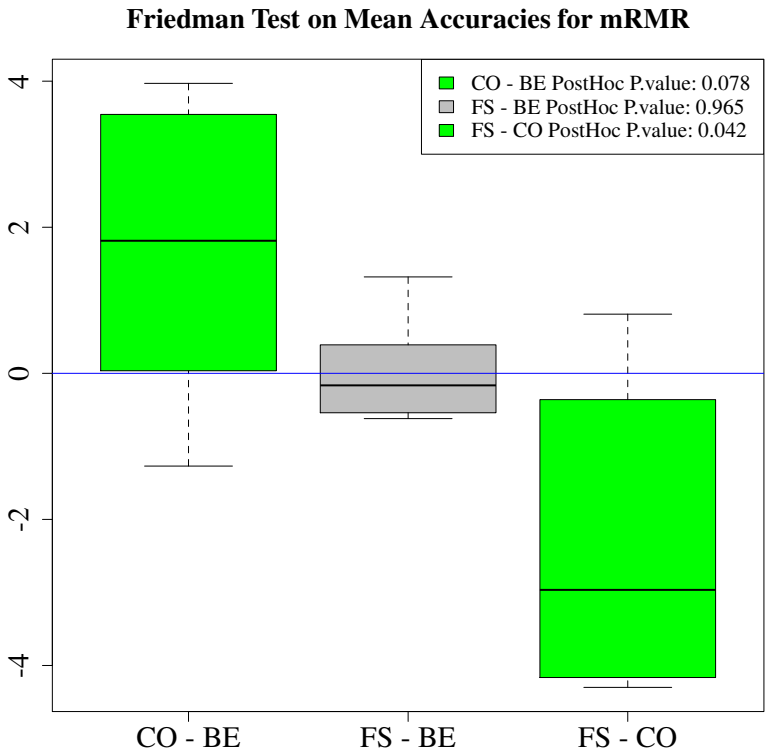

Figure 3.2: Comparing the search strategies for mRMR measure with the Friedman test and its corresponding post-hoc analysis. The Y-axis is the classification accuracy difference and X-axis indicates the names of the compared algorithms.

small datasets the estimated mutual information terms are highly inaccurate, features that rank low with our noisy measure function may in fact be better for classification. As seen in Table 3.3, for NCI the averaged similarity ratio is significantly smaller than 1 while for CNA which is a relatively larger dataset, it is almost constant and equal to 1.

The second interesting point is with respect to the MAD dataset. As can be seen, mRMR with greedy search algorithms perform poorly on this dataset.

Several authors have already utilized this dataset to compare their proposed criterion with mRMR and arrived at the conclusion that mRMR cannot handle highly correlated features, as in MAD dataset. However, surprisingly the performance of the mRMR+COBRA is as good as JMI on this dataset meaning that it is not the criterion but the search method that has difficulty to deal with highly correlated features. Thus, any conclusion with respect to the quality of a measure has to be drawn carefully since, as in this case, the effect of the non optimum search method can be decisive.

To discover the statistically significant differences between the algorithms, we applied the Friedman test following with Wilcoxon-Nemenyi post-hoc analysis, as suggested in [HW99], on the average accuracies (the last column of Tables 3.4 and 3.5). Note that since 8 datasets were used in the experiments, there are 8 independent measurements available for each algorithm. The results of this test for mRMR based algorithms have been depicted in Figure 3.2. In all box plots, CO stands for COBRA algorithm. Each box plot compares a pair of the algorithms. The green box plots represent the existence of a significant difference between the corresponding algorithms. The adjusted p-values for each pair of algorithms have also been reported in Figure 3.2. The smaller the p-value, the stronger the evidence against the null hypothesis. As can be seen, COBRA shows meaningful superiority over both greedy algorithms. However, if the significance level is set at $p = 0.05$, only FS rejects the null hypothesis and shows a meaningful difference with COBRA.

The same test was run for each classifier and its results can be found in Figure 3.3. While three of the classifiers show some differences between FS and COBRA, neither of them reveal any meaningful difference between BE and COBRA. At this point, the least one can conclude is that independent of the classification algorithm at hand, it is a good chance that FS performs worse than COBRA.

For JMI, however, the performances of all algorithms are comparable and with only 8 datasets it is difficult to draw any conclusion (see Figure 3.4).

In the next experiment COBRA is compared with two other convex programming based feature selection algorithms, SOSS [NHP13] and QPFS [RHEC10]. Both SOSS and QPFS employ quadratic programing techniques to maximize a score function. SOSS, however, uses an instance of randomized rounding to generate the set-membership binary values while QPFS ranks the features based on their scores (achieved from solving the convex problem) and therefore, sidesteps the difficulties of generating binary values. Note that

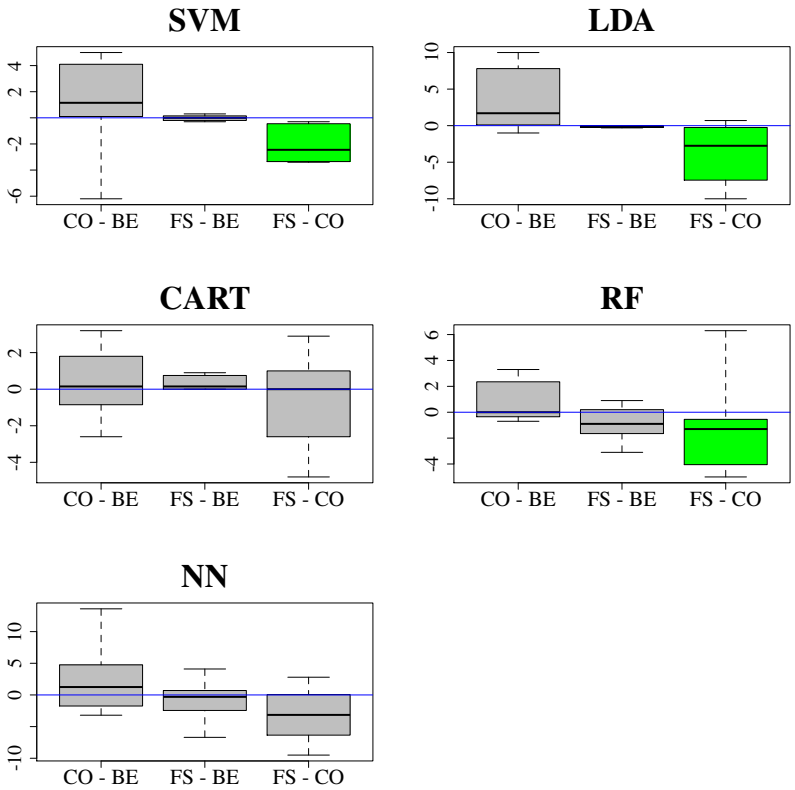

Figure 3.3: Comparing the search strategies for mRMR. Results of the post-hoc tests for each classifier.

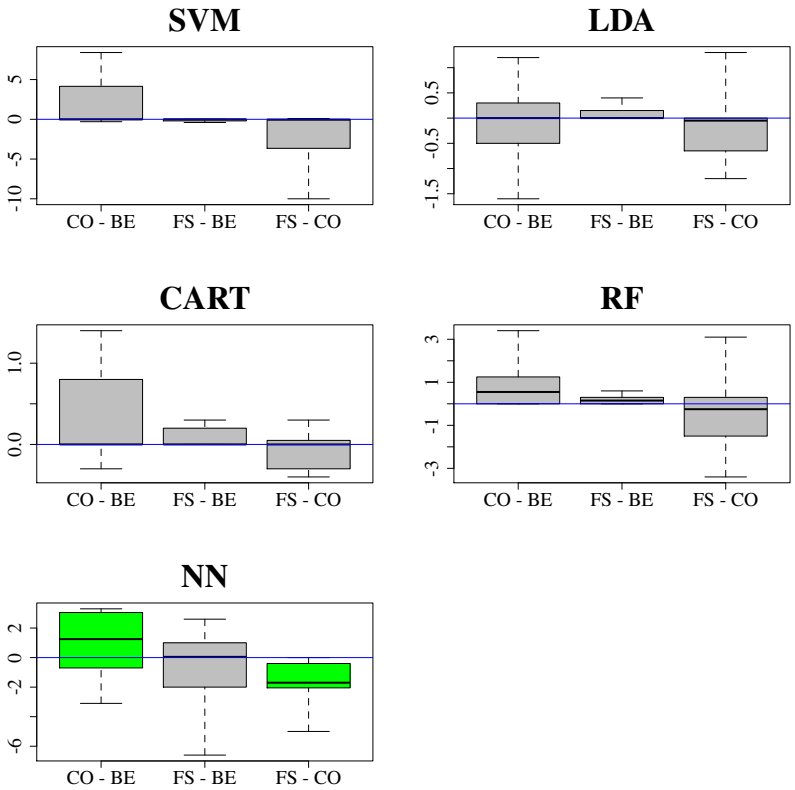

Figure 3.4: Comparing the search strategies for JMI. Results of the post-hoc tests for each classifier.

Datasets	mRMR+COBRA	mRMR+QPFS	mRMR+SOSS
MAD	74.81 ± 0.65	71.44 ± 0.57	71.36 ± 0.53
NCI	73.67 ± 2.41	71.00 ± 1.84	72.65 ± 2.13
IAD	96.64 ± 0.16	95.02 ± 0.21	96.64 ± 0.28
ARR	77.75 ± 1.03	78.73 ± 0.84	79.86 ± 1.18
CNA	88.91 ± 0.31	86.93 ± 0.45	85.43 ± 0.49

Table 3.6: Comparison of COBRA with QPFS and SOSS over 5 datasets. Average classification rates and their standard deviations are reported.

Datasets	Time COBRA	Time QPFS	Time SOSS
MAD	$175 + 24$	11	$175 + 5$
NCI	$368 + 341$	180	$368 + 27$
IAD	$540 + 121$	202	$540 + 12$
ARR	$6 + 14$	1	$6 + 4$
CNA	$120 + 50$	25	$120 + 7$

Table 3.7: Comparison of COBRA with QPFS and SOSS over 5 datasets. The computational times in second are reported where the first value for COBRA and SOSS is for calculating the mutual information matrix and the second value is the time required to solve the optimization problems.

both COBRA and SOSS first need to calculate the mutual information matrix $\mathbf{Q}$. Once it is calculated, they can solve their corresponding convex optimization problems for different values of P. Table 3.6 reports the average (over 5 classifiers) classification accuracies of these three algorithms and the standard deviation of these mean accuracies (calculated over the cross-validation folds). In Table 3.7, the computational times of each algorithm for a single run (in second) are shown, i.e., the amount of time required to select a feature set with (given) P features. The reported times for COBRA and SOSS consist of two values. The first value is the time required to calculate the mutual information matrix $\mathbf{Q}$ and the second value is the amount of time required to solve the corresponding convex optimization problem. All the values were measured on a PC with an Intel Core i7 CPU. As can be seen from Table 3.7, QPFS is significantly faster than COBRA and SOSS. This computational

Datasets		SVM	CART	RF	NN
ARR	LDA feat.	78.4	73.7	77.1	68.00
	Optimum	81.9	75.4	82.2	72.9
CNA	LDA feat.	92.6	75.0	90.5	91.1
	Optimum	94.0	75.0	90.8	92.0
IAD	LDA feat	95.8	96.0	97.2	96.3
	Optimum	96.5	96.4	97.2	97.1

Table 3.8: The performance of the classification algorithms when trained with COBRA features optimized for the LDA classifier. This table shows the generalization power of the COBRA features on the classifiers.

superiority, however, seems to come at the expense of lower classification accuracy. For large datasets such as IAD, CNA and MAD, the Nyström approximation used in QPFS to cast the problem into a lower dimensional subspace does not yield a precise enough approximation and results in lower classification accuracies. An important remark to interpret these results is that, for NCI dataset (in all the experiments) first features with the low mutual information values with the class label were filtered out and only 2000 informative features were kept simply because computing a mutual information matrix of size 9703×9703 was a computationally demanding task. Thus, the dimension is 2000 and not 9703 as mentioned in Table 3.2.

The generalization power of the COBRA algorithm over different classifiers is another important issue to test. As can be observed in Table 3.4, the number of selected features varies quite markedly from one classifier to another. However, based on our experiments, the optimum feature set of any of the classifiers, usually (for large enough datasets) achieves a near-optimal accuracy in conjunction with other classifiers as well. This is shown in Table 3.8 for 4 classifiers and 3 datasets. The COBRA features of the LDA classifier in Table 3.4 is used here to train other classifiers. Table 3.8 lists the accuracies obtained by using the LDA features and the optimal features, repeated from Table 3.4. Unlike the CNA and IAD datasets, a significant accuracy reduction can be observed in the case of ARR data which has substantially less training data than CNA and IAD. It suggests that for small size datasets, a feature selection scheme should take the induction algorithm into account since the learning algorithm is sensitive to small changes of the feature set.

3.5 COBRA-selected ISCN Features for Lipreading

Having the COBRA algorithm at hand, we can train our first visual speech recognizer with ISCN features described in Chapter 2. This recognizer is based on the GMM-HMM technology. First, the COBRA feature selection algorithm was used to select 74 features from the large set of ISCN features extracted from each frame. The first-order derivatives of these features were then included in this feature set. Compared with conventional visual features such as PCA, LDA and DCT features, COBRA selected ISCN features resulted in a visual speech recognizer with 6% absolute recognition rate improvement and 12% lower variance of accuracy over different speakers.

In this experiment the CUAVE dataset [PGTG02] is used which contains the digits from zero to nine repeated five times by 36 speakers. A brief description of this dataset is given in Chapter 2.2.1.

The size of the ROIs in CUAVE dataset are 128x128 pixels. In these experiments, 4 sets of features were extracted from these ROIs and compared.

1. ISCN: First and second order ISCN features were extracted from ROIs. By using a 3-layer network and setting the hyper-parameter J corresponding to the spatial resolution to three, about 50000 ISCN features were computed for the ROI of each video frame. Due to the high-dimensionality of the ISCN features, they cannot directly be used in the statistical modeling. Hence, the COBRA algorithm was employed to reduce the dimensionality by selecting a small subset of ISCN features.

2. PCA: The commonly used principal component analysis was applied to ROIs. The derived principal component scores corresponding to the largest eigenvalues were then considered the PCA features.

3. LDA: Linear discriminate analysis is performed on ROIs. The projection of the ROIs to a low dimensional linear subspace maximizing the projected ratio between the between-class scatter matrix and within-class scatter matrix yielded the LDA features.

4. DCT: The two dimensional DCT transform was applied to ROIs and features with the highest energy were selected in the zig-zag order so that the higher energy features appear first in the feature vector.

The first order temporal derivatives of the features are also included in all the feature sets.

By applying a two dimensional DCT to ISCN features and dropping 80% of the low-energy coefficients the feature space dimensionality was dramatically reduced. However, even keeping only 20% of the features resulted in 10000 features which is still prohibitively large. Hence, COBRA was used to reduce the feature set cardinality to a practically workable number.

Two set of experiments, one with word models and one with viseme models, were performed. Visemes are the visual equivalent of phonemes and are used to describe articulatory gestures in lipreading. A correspondence between visemes and phonemes are shown in Table 3.10. This map was suggested by Jeffers and Barley [JB71] to group 43 phonemes into 11 visemes. It was shown in [CH23] that compared with other viseme-phoneme maps, this correspondence achieves higher recognition accuracy in visual speech recognition. When visemes were used as speech units, 10 three-state HMMs were trained to statistically describe the visemes (10 visemes were sufficient to describe the digits). Each markov state was modeled with a GMM containing two mixture components with diagonal covariance matrices. When words were directly modeled, an HMM with 9 emitting states (two Gaussian mixture components with diagonal covariance matrix per state) was trained for each digit.

In the first set of experiments the performance of ISCN features were evaluated in the case where speech units are words. The visual-only speech recognizer was trained with various number of ISCN features selected by COBRA. In order to have a speaker independent evaluation, the leave one speaker out cross validation strategy was utilized. Figure 3.5 shows the recognition rates of different feature sets for test with word HMMs. As can be seen the graph corresponding to the COBRA-selected ISCN features is not smooth due to the randomization step in the COBRA algorithm, which results in selecting very diverse feature sets. However, the general trend is clear. By increasing the number of features the recognition rate improves and reaches 75% for a feature set with 148 features including the first order derivatives. After passing that optimal point, it starts to decrease because of the curse of dimensionality effect and ends up at around 72.6% recognition rate for 220 features. The performances of the V-ASR for LDA, PCA and DCT with respect to the number of features have also been depicted in Figure 3.5.

COBRA selected ISCN features considerably outperform the conventional visual features. Note that both ISCN and LDA features are selected

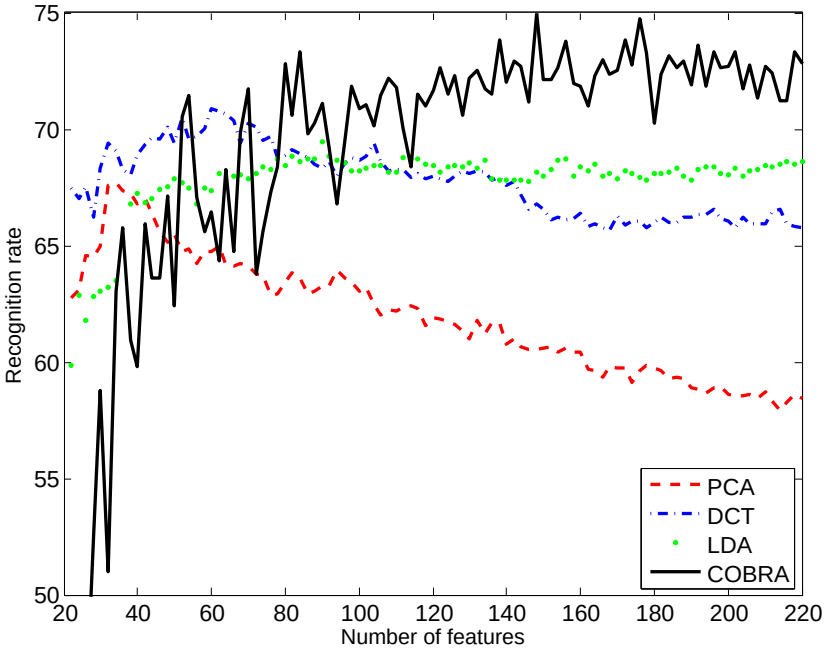

Figure 3.5: Comparing ISCN features selected by COBRA with conventional features in a digit recognition task with word models.

in a supervised way to maximize the viseme recognition rate. However, in the first experiment, the selected features were used to model the whole words rather than their constituent visemes. Table 3.9 reports the accuracy rates for various feature sets when the optimal number of features were used to train the V-ASR. The standard deviations of the reported mean accuracies were computed as the square root of the variance of the accuracy rates over the 36 speakers divided by $\sqrt{36}$ and is an indication of the sensitivity of the recognizer to speaker variation. Two other sensitivity indicators reported for each feature set are the mean accuracies of the top and bottom 10 percent of the recognition rates of the 36 speakers. The standard deviations reported in Table 3.9 shows that using ISCN features leads to about 12% relative variance reduction compared with LDA features. However, the gap between the performances of the best and worst groups (i.e., 10% Max and 10% Min) reveals

Feature types	# Features	Accuracy rates	10% Max	10% Min
ISCN	148	75%±2.99%	98.00	36.50
LDA	90	69%±3.38%	96.00	29.00
PCA	34	67%±2.91%	89.77	35.74
DCT	62	70%±2.58%	89.30	40.37

Table 3.9: Digit recognition rates and their corresponding standard deviations, for multiple feature sets.

that ISCN was not so successful in reducing this gap. This large gap between the maximum and minimum performances can be interpreted as the reliability of the V-ASR. As it is seen later, to have a successful audio-visual information fusion the V-ASR reliability should be improved. As mentioned, LDA selects a set of transformed features that has the largest ratio of between-class scatter to within-class scatter. Basically, LDA tends to maximize the separability by increasing the difference between the class-conditional mean values while keeping the within-class variances small. LDA is the optimum feature transformation method when class conditional distributions are Gaussian with similar covariance matrices and different mean values.

As can be seen, the highest recognition rate that LDA feature can achieve is about 69% with 90 features. The fact that LDA features cannot perform as well as ISCN features is an indication of non-Gaussianity of the underlying distribution. Since only two Gaussian terms in GMMs were used to model the in-state features distributions, it is not surprising that non-Gaussian distributions may not be accurately represented. Note that considering the limited available data, increasing the number of Gaussian components leads to performance degradation due to the excessive overfitting effect.

In the second set of experiments the performance of the COBRA selected ISCN features were evaluated when a sequence of viseme models were used to describe a digit. A correspondence between visemes and phonemes and the visemic pronunciations of digits are listed in Tables 3.10 and 3.11, respectively. As in the previous experiment, various number of features were used to train the visual speech recognizer. Figure 3.6 shows the recognition rates of different feature sets for the isolated digit recognition test. The ISCN features again outperform other feature sets however, the recognition rate is about 20% lower than that of word-based V-ASR reported in Table 3.9. This in fact can be explained if the video data are reviewed. In many occasions, the

Visemes	Visibility Rank	Occurrence%	TIMIT phonemes
/A	1	3.15	/f/ /v/
/B	2	15.49	/er/ /ow/ /r/ /q/ /w/ /uh/ /uw/ /axr/ /ux/
/C	3	5.88	/b/ /p/ /m/ /em/
/D	4	0.70	/aw/
/E	5	2.90	/dh/ /th/
/F	6	1.20	/ch/ /jh/ /sh/ /zh/
/G	7	1.81	/oy/ /ao/
/H	8	4.36	/s/ /z/
/I	9	31.46	/aa/ /ae/ /ah/ /ay/ /eh/ /ey/ /ih/ /iy/ /y/ /ao/ /ax-h/ /ax/ /ix/
/J	10	21.10	/d/ /l/ /n/ /t/ /el/ /nx/ /en/ /dx/
/K	11	4.84	/g/ /k/ /ng/ /eng/
/S	-	-	/sil/

Table 3.10: The visemes and their corresponding phonemes suggested in [JB71] and evaluated in [CH23]. The TIMIT phonetic alphabet notation is used to describe the phonemes (see [Hie93] for mapping TIMIT to IPA). The visibility rank is an indication of the difficulty of recognition. The lower this number is, the more difficult it is to recognize a viseme.

Digit	Pronunciation	Digit	Pronunciation
zero	[H + I + B + B]	one	[B + I + J]
two	[J + B]	three	[E + B + I]
four	[A + G + B]	five	[A + I + A]
six	[H + I + K + H]	seven	[H + I + A + I + J]
eight	[I + J]	nine	[J + I + J]

Table 3.11: The viseme transcriptions of digits.

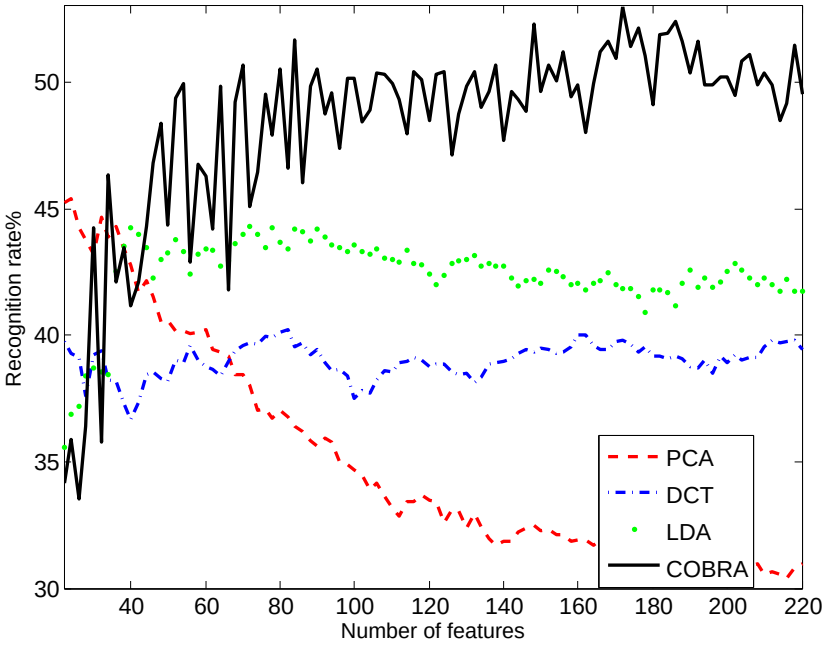

Figure 3.6: Comparing ISCN features selected by COBRA with conventional features in a digit recognition task with viseme models.

visemes are not even pronounced (for instance, /t/ at the end of the digit eight is very frequently omitted) or when pronounced, they only appear in one to three frames which is too short for modeling the time-dynamics of them.

Figure 3.7 reports the confusion matrix of the digit classification when visemes were used to describe the digits.

The ground truth can be found in the rows of the confusion matrix while the columns are devoted to predicted labels. Most confusions in digit recognitions are caused by the phonemes of two words being mapped to a same viseme. However, it is also possible that different visemes have the same visual appearance depending on the context. This type of confusion is caused by a phenomena commonly called co-articulation [CM93] where the mouth shape of a particular phoneme causes the next phonemes to have similar mouth shapes. For instance, in the words *eight* and *nine*, the viseme /I is the

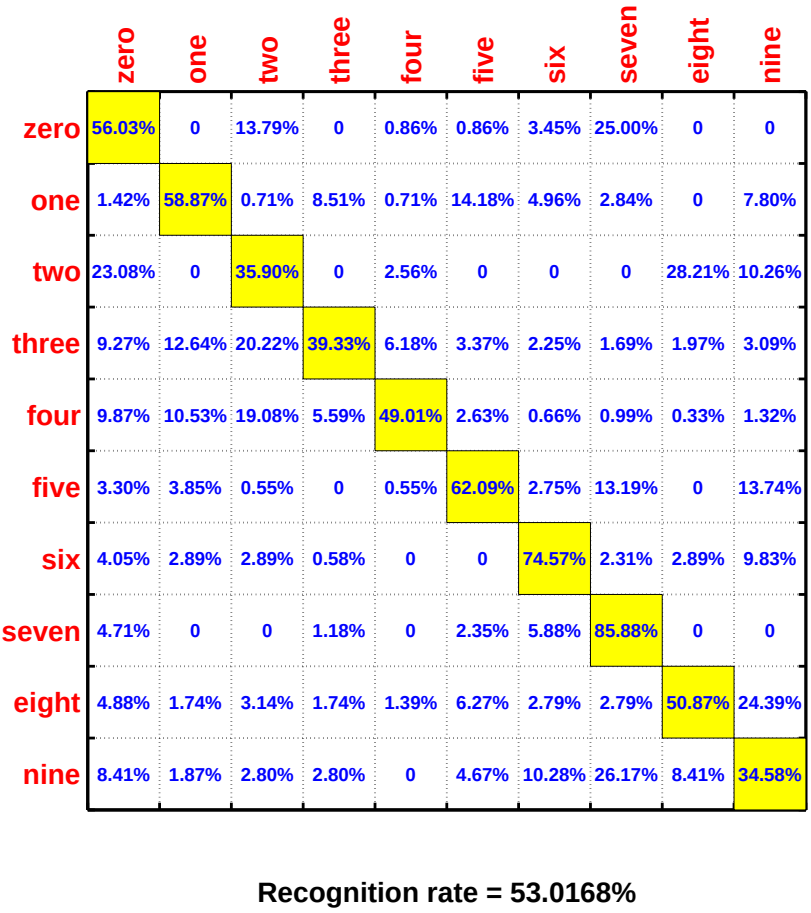

Figure 3.7: Confusion matrix of digit recognition with visemes. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class.

dominant viseme governing the mouth shape. That is, in the presence of /I, the viseme /J in *eight* ([I + J]) and in *nine* ([J + I + J]) does not have a visible effect.

Even though, there is a drastic performance reduction compared with

when word models are employed, the classification results with visemes are far from being random. Given better models for visemes, it is plausible to achieve higher performances.

3.6 Conclusion and Discussion

In this chapter, a mutual information based feature selection algorithm was developed that can select a close-to-optimal subset of informative features. It was shown that by using a semi-definite programming based search strategy, this problem can be seen as a variant of the maximum-cut problem in graph theory. This observation was the basis of our approximation ratio analysis. The approximation ratio of COBRA was derived and compared with the approximation ratio of the backward elimination method. It was experimentally shown that COBRA outperforms sequential search methods especially in the case of sparse data.

Two series expansions for mutual information were represented and was shown that most mutual information based score functions in the literature including mRMR and MIFS are truncated approximations of these expansions. Furthermore, the underlying connection between MIFS and the Kirwood approximation was explored, and it was shown that by adopting the class conditional independence assumption and the Kirkwood approximation for $Pr(\mathbf{X})$, mutual information reduces to the MIFS criterion.

This algorithm is employed to select a subset of ISCN features and used these features to train a GMM-HMM based speech recognizer. As shown in the experiments, using these features leads to 5-6% accuracy improvement over the commonly used features derived from PCA, LDA and DCT transformations. Using ISCN, however, does not solve the reliability problem associated with the V-ASR. If the V-ASR performance had an acceptable variance over different speakers, it could be directly fused with audio recognizer by multiplying their posterior probabilities. However, due to its large inter-speaker variance a more complex fusion scheme is required to properly weight the modalities according to their reliabilities. This issue is discussed in Chapter 7.

Chapter 4

Binary and Multiclass Classification

A learning machine or an inducer is an algorithm that induces a hypothesis to describe the relation between features and labels from a training set. The main focus of this chapter is on dealing with classification problems. Therefore, the terms learning machine and classifier are used interchangeably.

In many applications simple classifiers such as linear SVM or GMM-based naive Bayes classifiers are sufficient to obtain a high level of accuracy. In lipreading, however, this is not the case due to the complex nature of the underlying random process. Most learning algorithms applied to this problem underfit the complex reality and result in lower-than-expected accuracy. To address this issue, a powerful class of learners based on boosting methods, which are known to be efficient[1] learning algorithms, is introduced. In this approach, a large number of weak classifiers are generated and the final hypothesis is taken to be the convex combination of these weak-classifiers. First a framework for the binary classification setting is provided and is shown that there is a natural generalization of it to multiclass setting. The robustness of

[1] An algorithm is called efficient if (I) it can find the optimal hypothesis in time polynomial in the length of the input and the number of samples (this part related to the training phase) and (II) it can compute the value of the optimal hypothesis in time polynomial in the input dimensionality at the test phase. For instance, the K-nearest neighbors algorithm is not an efficient algorithm due to the fact that its computational complexity at the test phase goes to infinity as the number of samples approaches infinity.

the proposed classifiers against overfitting and mislabeling noise is of great help in constructing robust voice activity detection and isolated word lipreading systems.

4.1 Introduction to Boosting Problem

A boosting algorithm can be seen as a meta-algorithm that maintains a distribution over the sample space. At each iteration a weak hypothesis is learned and the distribution is updated, accordingly. The output (strong hypothesis) is a convex combination of the weak hypotheses. Two dominant views to describe and design boosting algorithms are "weak to strong learner" (WTSL), which is the original viewpoint presented in [Sch90, FS97], and boosting by "coordinate-wise gradient descent in the functional space" (CWGD) appearing in later works [Bre99, MBBF99, FHT98]. A boosting algorithm adhering to the first view guarantees that it only requires a finite number of iterations (equivalently, finite number of weak hypotheses) to learn a $(1-\epsilon)$-accurate hypothesis. In contrast, an algorithm resulting from the CWGD viewpoint (usually called potential booster) may not necessarily be a boosting algorithm in the probably approximately correct (PAC) learning sense. However, while it is rather difficult to construct a boosting algorithm based on the first view, the algorithmic frameworks, e.g., AnyBoost [MBBF99], resulting from the second viewpoint have proven to be particularly prolific for developing new boosting algorithms. Under the CWGD view, the choice of the convex loss function to be minimized is the cornerstone of designing a boosting algorithm. This, however, is a severe disadvantage in some applications.

In CWGD, the weights are not directly controllable (designable) and are only viewed as the values of the gradient of the loss function. In many applications, some characteristics of the desired distribution are known or given as problem requirements while, finding a loss function that generates such a distribution is likely to be difficult. For instance, what loss functions can generate sparse distributions?[2] What family of loss functions results in a smooth distribution?[3] We even can go further and imagine the scenarios in which a

[2] In the boosting terminology, sparsity usually refers to the greedy hypothesis-selection strategy of boosting methods in the functional space. However, sparsity in this chapter refers to the sparsity of the distribution (weights) over the sample space.

[3] A smooth distribution is a distribution that does not put too much weight on any single sample or in other words, a distribution emulated by the booster does not dramatically diverge from the target distribution [Ser03, Gav03].

loss function needs to put more weights on a given subset of examples than others, either because that subset has more reliable labels or it is a problem requirement to have a more accurate hypothesis for that part of the sample space. Then, what loss function can generate such a customized distribution? Moreover, does it result in a provable boosting algorithm? In general, how can we characterize the accuracy of the final hypothesis?

Although, to be fair, the so-called loss function hunting approach has given rise to useful boosting algorithms such as LogitBoost, FilterBoost, GiniBoost and MADABoost [FHT98, BS08, Hat06, DW00] which (to some extent) answer some of the above questions, it is an inflexible and relatively unsuccessful approach to addressing the boosting problems with distribution constraints.

Another approach to designing a boosting algorithm is to directly follow the WTSL viewpoint [Fre95, Ser03, BGL02]. The immediate advantages of such an approach are, first, the resultant algorithms are provable boosting algorithms, i.e., they output a hypothesis of arbitrary accuracy. Second, the booster has direct control over the weights, making it more suitable for boosting problems subject to some distribution constraints. However, since the WTSL view does not offer any algorithmic framework (as opposed to the CWGD view), it is rather difficult to come up with a distribution update mechanism resulting in a provable boosting algorithm. There are, however, a few useful, and albeit fairly limited, algorithmic frameworks such as TotalBoost [WLR06] that can be used to derive other provable boosting algorithms. The TotalBoost algorithm can maximize the margin by iteratively solving a convex problem with the totally corrective constraint. A more general family of boosting algorithms was later proposed by Shalev-Shwartz et al. [SS08], where it was shown that weak learnability and linear separability are equivalent, a result following from von Neumann's minmax theorem. Using this theorem, they constructed a family of algorithms that maintain smooth distributions over the sample space, and consequently are noise tolerant. Their proposed algorithms find an $(1-\epsilon)$-accurate solution after performing at most $O(\log(N)/\epsilon^2)$ iterations, where N is the number of training examples.

4.2 Our Results

A family of boosting algorithms is presented that can be derived from well-known online learning algorithms, including projected gradient descent

[Zin03] and its generalization, mirror descent (both active and lazy updates, see [Haz09]) and composite objective mirror descent (COMID) [DSST10]. The PAC learnability of the algorithms derived from this framework is proven and shown that this framework in fact generates maximum margin algorithms. That is, given a desired accuracy level ν, it outputs a hypothesis of margin $\gamma_{\text{min}} - \nu$ with γ_{min} being the minimum edge that the weak classifier guarantees to return.

The duality between (linear) online learning and boosting is by no means new. In online learning at round t, the learner receives a new unlabeled sample point and is required to predict its label. After predicting a label, the correct label is revealed and the learner suffers some amount of loss (dependent on the loss function). In boosting, however, at each time instance t, all samples are available to the booster. The booster selects a weak learner from the hypothesis space and suffers some amount of loss due to the performance of the selected weak learner. Both of these methods can be seen as zero-sum games with very similar formulations. The duality between these two methods was first pointed out in [FS97] and was later elaborated and formalized by using the von Neumann's minmax theorem [FS96b].

Following this line, several proof techniques required to show the PAC learnability of the derived boosting algorithms are provided. These techniques are fairly versatile and can be used to translate many other online learning methods into our boosting framework. To motivate our boosting framework, two practically and theoretically interesting algorithms are derived:

- The **SparseBoost algorithm** which by maintaining a sparse distribution over the sample space tries to reduce the space and the computation complexity. In fact this problem, i.e., applying batch boosting on the successive subsets of data when there is not sufficient memory to store an entire dataset, was first discussed by Breiman in [Bre97], though no algorithm with theoretical guarantee was suggested. SparseBoost is the first provable batch booster that can (partially) address this problem. By analyzing this algorithm, it is shown that the tuning parameter of the regularization term ℓ_1 at each round t should not exceed $\frac{\gamma_t}{2}\eta_t$ to still have a boosting algorithm, where η_t is the coefficient of the t^{th} weak hypothesis and γ_t is its edge. Exploiting sparsity in example domain has also been investigated in [HT09] by Hatano and Takimoto.

- A **smooth boosting algorithm** that requires only $O(\log 1/\epsilon)$ number of rounds to learn a $(1-\epsilon)$-accurate hypothesis. This algorithm can

also be seen as an agnostic boosting algorithm[4] due to the fact that smooth distributions provide a theoretical guarantee for noise tolerance in various noisy learning settings, such as agnostic boosting [KK09, BLM01].

Furthermore, an interesting theoretical result about MADABoost [DW00] is provided. A proof (to the best of our knowledge the only available unconditional proof) for the boosting property of (a variant of) MADABoost is given here and is shown that, unlike the common presumption, its convergence rate is of $O(1/\epsilon^2)$ rather than $O(1/\epsilon)$.

The MABoost framework to multiclass setting is generalized and shown that it adopts the minimal weak-learning condition introduced in [MS13]. That is, it imposes minimal conditions on the weak learner space to drive the training error to zero. The ADABoost.MM algorithm presented in [MS13] can also be derived from our framework by using the Kullback-Leibler (KL) divergence for the generic Bregman divergence in the MABoost framework.

4.3 Fundamentals

First of all, the notations used throughout the chapter is established. Vectors are lower case bold letters and their entries are non-bold letters with subscripts, such as x_i of $\mathbf{x}$, or non-bold letter with superscripts if the vector already has a subscript, such as x_t^i of $\mathbf{x}_t$. Matrices are upper case bold letters and their entries are shown by upper case non-bold letters with subscripts. Moreover, an N-dimensional probability simplex is denoted by $\mathcal{S} = \{\mathbf{w} | \sum_{i=1}^{N} w_i = 1, w_i \geq 0\}$.

Since a central notion throughout this chapter is that of Bregman divergences, some of their properties are briefly revisited. A Bregman divergence is defined with respect to a convex function $\mathcal{R}$ as

$$B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) = \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{y}) - \nabla\mathcal{R}(\mathbf{y})(\mathbf{x} - \mathbf{y})^{\top} \tag{4.1}$$

and can be interpreted as a distance measure between $\mathbf{x}$ and $\mathbf{y}$. Due to the convexity of $\mathcal{R}$, a Bregman divergence is always non-negative, i.e.,

[4] Unlike the PAC model, the agnostic learning model allows an arbitrary target function (labeling function) that may not belong to the class studied, and hence, can be viewed as a noise tolerant learning model [KSS92].

$B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) \geq 0$. In this work $\mathcal{R}$ is considered to be a β-strongly convex function[5] with respect to a norm $||.||$. With this choice of $\mathcal{R}$, the Bregman divergence $B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) \geq \frac{\beta}{2}||\mathbf{x} - \mathbf{y}||^2$. As an example, if $\mathcal{R}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{x}$ (which is 1-strongly convex with respect to $||.||_2$), then $B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) = \frac{1}{2}||\mathbf{x} - \mathbf{y}||_2^2$ is the square of the Euclidean distance. Another example is the negative entropy function $\mathcal{R}(\mathbf{x}) = \sum_{i=1}^{N} x_i \log x_i$ (resulting in the KL-divergence) which is known to be 1-strongly convex over the probability simplex with respect to ℓ_1 norm.

The Bregman projection is another fundamental concept of our framework.

Definition 4.1. Bregman Projection *The Bregman projection of a vector* $\mathbf{y}$ *onto a convex set* $\mathcal{S}$ *with respect to a Bregman divergence* $B_{\mathcal{R}}$ *is*

$$\Pi_{\mathcal{S}}(\mathbf{y}) = \arg\min_{\mathbf{x} \in \mathcal{S}} B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) \tag{4.2}$$

Moreover, the following generalized Pythagorean theorem holds for Bregman projections.

Lemma 4.2. Generalized Pythagorean (see also [CBL06, Lemma 11.3]) *Given a point* $\mathbf{y} \in \mathbb{R}^N$*, a convex set* $\mathcal{S}$ *and* $\hat{\mathbf{y}} = \Pi_{\mathcal{S}}(\mathbf{y})$ *as the Bregman projection of* $\mathbf{y}$ *onto* $\mathcal{S}$*, for all* $\mathbf{x} \in \mathcal{S}$ *we have*

$$\textit{Exact:} \qquad B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) \geq B_{\mathcal{R}}(\mathbf{x}, \hat{\mathbf{y}}) + B_{\mathcal{R}}(\hat{\mathbf{y}}, \mathbf{y}) \tag{4.3}$$

$$\textit{Relaxed:} \qquad B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) \geq B_{\mathcal{R}}(\mathbf{x}, \hat{\mathbf{y}}) \tag{4.4}$$

The relaxed version follows from the fact that $B_{\mathcal{R}}(\hat{\mathbf{y}}, \mathbf{y}) \geq 0$ and thus can be ignored.

Lemma 4.3. *For any vectors* $\mathbf{x}, \mathbf{y}, \mathbf{z}$*, we have*

$$(\mathbf{x} - \mathbf{y})^\top (\nabla \mathcal{R}(\mathbf{z}) - \nabla \mathcal{R}(\mathbf{y})) = B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) - B_{\mathcal{R}}(\mathbf{x}, \mathbf{z}) + B_{\mathcal{R}}(\mathbf{y}, \mathbf{z}) \tag{4.5}$$

The above lemma follows directly from the Bregman divergence definition in (4.1). Additionally, the following definitions from convex analysis are useful throughout the chapter.

[5] That is, its second derivative (Hessian in higher dimensions) is bounded away from zero by at least β.

Definition 4.4. Norm & dual norm *Let $||.||_A$ be a norm. Then its dual norm is defined as*

$$||\mathbf{y}||_{A^*} = \sup\{\mathbf{y}^\top \mathbf{x}, ||\mathbf{x}||_A \leq 1\} \tag{4.6}$$

For instance, the dual norm of $||.||_2 = \ell_2$ is $||.||_{2^*} = \ell_2$ norm and the dual norm of ℓ_1 is ℓ_∞ norm. Further,

Lemma 4.5. *For any vectors $\mathbf{x}, \mathbf{y}$ and any norm $||.||_A$, the following inequality holds:*

$$\mathbf{x}^\top \mathbf{y} \leq ||\mathbf{x}||_A ||\mathbf{y}||_{A^*} \leq \frac{1}{2}||\mathbf{x}||_A^2 + \frac{1}{2}||\mathbf{y}||_{A^*}^2 \tag{4.7}$$

The proof directly follows from Hölder's inequality. Throughout this chapter, the shorthands $||.||_A = ||.||$ and $||.||_{A^*} = ||.||_*$ are used for the norm and its dual, respectively.

4.4 Boosting Framework

Let $\{(\mathbf{x}_i, a_i)\}, 1 \leq i \leq N$, be N training samples, where $\mathbf{x}_i \in \mathcal{X}$ and $a_i \in \{-1, +1\}$. Assume $h \in \mathcal{H}$ is a real-valued function mapping $\mathcal{X}$ into $[-1, 1]$. Let us denote a distribution over the training data by $\mathbf{w} = [w_1, \ldots, w_N]^\top$ and define a loss vector $\mathbf{d} = [-a_1 h(\mathbf{x}_1), \ldots, -a_N h(\mathbf{x}_N)]^\top$. We define $\gamma = -\mathbf{w}^\top \mathbf{d}$ as the *edge* of the hypothesis h under the distribution w and it is assumed to be positive when h is returned by a weak learner. In this work, the branching program based boosters introduced in [MM02] is not considered and we adhere to the typical boosting protocol (described in Section 4.1).

Let $\mathcal{R}(\mathbf{x})$ be a 1-strongly convex function with respect to a norm $||.||$ and denote its associated Bregman divergence $B_\mathcal{R}$. Moreover, let the dual norm of a loss vector $\mathbf{d}_t$ be upper bounded, i.e., $||\mathbf{d}_t||_* \leq L$. The following mirror ascent boosting (MABoost) algorithm is our boosting framework. It is easy to verify that for $\mathbf{d}_t$ as defined in MABoost, $L = 1$ when $||.||_* = \ell_\infty$ and $L = N$ when $||.||_* = \ell_2$.

Algorithm 4.1: Mirror Ascent Boosting (MABoost)

Input: $\mathcal{R}(\mathbf{x})$ 1-strongly convex function,
$\mathbf{w}_1 = [\frac{1}{N}, \ldots, \frac{1}{N}]^\top$ and $\mathbf{z}_1 = [\frac{1}{N}, \ldots, \frac{1}{N}]^\top$

For $t = 1, \ldots, T$ **do**

(a) Train classifier with $\mathbf{w}_t$ and get h_t,
let $\mathbf{d}_t = [-a_1 h_t(\mathbf{x}_1), \ldots, -a_N h_t(\mathbf{x}_N)]$ and $\gamma_t = -\mathbf{w}_t^\top \mathbf{d}_t$.

(b) Set $\eta_t = \frac{\gamma_t}{L}$

(c) Update weights: $\nabla\mathcal{R}(\mathbf{z}_{t+1}) = \nabla\mathcal{R}(\mathbf{z}_t) + \eta_t \mathbf{d}_t$ (lazy update)
$\nabla\mathcal{R}(\mathbf{z}_{t+1}) = \nabla\mathcal{R}(\mathbf{w}_t) + \eta_t \mathbf{d}_t$ (active update)

(d) Project onto $\mathcal{S}$: $\mathbf{w}_{t+1} = \operatorname{argmin}_{\mathbf{w}\in\mathcal{S}} B_{\mathcal{R}}(\mathbf{w}, \mathbf{z}_{t+1})$

End

Output: The final hypothesis $f(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=1}^{T} \eta_t h_t(\mathbf{x})\right)$.

This algorithm is a variant of the mirror descent algorithm [Haz09], modified to work as a boosting algorithm. The basic principle in this algorithm is quite clear. As in ADABoost, the weight of a wrongly (correctly) classified sample increases (decreases). The weight vector is then projected onto the probability simplex in order to keep the weight sum equal to 1. The distinction between the active and lazy update versions and the fact that the algorithm may behave quite differently under different update strategies should be emphasized. In the lazy update version, the norm of the auxiliary variable $\mathbf{z}_t$ is unbounded. In the active update version, on the other hand, $\mathbf{z}$ is bounded and does not drive far away from $\mathbf{w}$. However, unlike the lazy version in the active update the algorithm always needs to access (compute) the previous projected weight $\mathbf{w}_t$ to update the weight at round t and this may not be possible in some applications (such as boosting-by- filtering [DW00]).

Due to the duality between online learning and boosting, it is not surprising that MABoost (both the active and lazy versions) is a boosting algorithm. The proof of its boostability, however, reveals some interesting properties which enables us to generalize the MABoost framework. In the following, only the proof of the active update is given and the lazy update is left to Section 4.4.4.

Theorem 4.6. *Suppose that MABoost generates weak hypotheses* $h_1, \ldots, h_T$

whose edges are $\gamma_1, \ldots, \gamma_T$. Then the error ϵ of the combined hypothesis f on the training set is bounded as:

$$\mathcal{R}(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||_2^2 : \qquad \epsilon \leq \frac{1}{1 + \sum_{t=1}^{T} \gamma_t^2} \tag{4.8}$$

$$\mathcal{R}(\mathbf{w}) = \sum_{i=1}^{N} w_i \log w_i : \qquad \epsilon \leq e^{-\sum_{t=1}^{T} \frac{1}{2}\gamma_t^2} \tag{4.9}$$

In fact, the first bound (4.8) holds for any 1-strongly convex $\mathcal{R}$, though for some $\mathcal{R}$ (e.g., negative entropy) the much tighter bound as in (4.9) can be achieved.

Proof: Assume $\mathbf{w}^* = [w_1^*, \ldots, w_N^*]^\top$ is a distribution vector where $w_i^* = \frac{1}{N\epsilon}$ if $f(\mathbf{x}_i) \neq a_i$, and 0 otherwise. $\mathbf{w}^*$ can be seen as a uniform distribution over the wrongly classified samples by the ensemble hypothesis f. Using this vector and following the approach in [Haz09], the upper bound of $\sum_{t=1}^{T} \eta_t(\mathbf{w}^{*\top}\mathbf{d}_t - \mathbf{w}_t^\top \mathbf{d}_t)$ is derived where $\mathbf{d}_t = [d_t^1, \ldots, d_t^N]$ is a loss vector as defined in Algorithm 4.1.

$$(\mathbf{w}^* - \mathbf{w}_t)^\top \eta_t \mathbf{d}_t = (\mathbf{w}^* - \mathbf{w}_t)^\top \big(\nabla\mathcal{R}(\mathbf{z}_{t+1}) - \nabla\mathcal{R}(\mathbf{w}_t)\big) \tag{4.10a}$$

$$= B_\mathcal{R}(\mathbf{w}^*, \mathbf{w}_t) - B_\mathcal{R}(\mathbf{w}^*, \mathbf{z}_{t+1}) + B_\mathcal{R}(\mathbf{w}_t, \mathbf{z}_{t+1}) \tag{4.10b}$$

$$\leq B_\mathcal{R}(\mathbf{w}^*, \mathbf{w}_t) - B_\mathcal{R}(\mathbf{w}^*, \mathbf{w}_{t+1}) + B_\mathcal{R}(\mathbf{w}_t, \mathbf{z}_{t+1}) \tag{4.10c}$$

where the first equation follows Lemma 4.3 and inequality (4.10c) results from the relaxed version of Lemma 4.2. Note that Lemma 4.2 can be applied here because $\mathbf{w}^* \in \mathcal{S}$.

Further, the $B_\mathcal{R}(\mathbf{w}_t, \mathbf{z}_{t+1})$ term is bounded. By applying Lemma 4.5

$$B_\mathcal{R}(\mathbf{w}_t, \mathbf{z}_{t+1}) + B_\mathcal{R}(\mathbf{z}_{t+1}, \mathbf{w}_t) = (\mathbf{z}_{t+1} - \mathbf{w}_t)^\top \eta_t \mathbf{d}_t$$
$$\leq \frac{1}{2}||\mathbf{z}_{t+1} - \mathbf{w}_t||^2 + \frac{1}{2}\eta_t^2 ||\mathbf{d}_t||_*^2 \tag{4.11}$$

and since $B_\mathcal{R}(\mathbf{z}_{t+1}, \mathbf{w}_t) \geq \frac{1}{2}||\mathbf{z}_{t+1} - \mathbf{w}_t||^2$ due to the 1-strongly convexity of $\mathcal{R}$, we have

$$B_\mathcal{R}(\mathbf{w}_t, \mathbf{z}_{t+1}) \leq \frac{1}{2}\eta_t^2 ||\mathbf{d}_t||_*^2 \tag{4.12}$$

Now, substituting (4.12) into (4.10c) and summing it up from $t = 1$ to T, yields

$$\sum_{t=1}^{T} \mathbf{w}^{*\top} \eta_t \mathbf{d}_t - \mathbf{w}_t^{\top} \eta_t \mathbf{d}_t \leq \sum_{t=1}^{T} \frac{1}{2} \eta_t^2 ||\mathbf{d}_t||_*^2 + B_{\mathcal{R}}(\mathbf{w}^*, \mathbf{w}_1) - B_{\mathcal{R}}(\mathbf{w}^*, \mathbf{w}_{T+1}) \tag{4.13}$$

Moreover, it is evident from the algorithm description that for wrongly classified samples

$$-a_i f(\mathbf{x}_i) = -a_i \text{sign}\left(\sum_{t=1}^{T} \eta_t h_t(\mathbf{x}_i) \right) = \text{sign}\left(\sum_{t=1}^{T} \eta_t d_t^i \right) \geq 0 \qquad \forall \mathbf{x}_i \in \{\mathbf{x} | f(\mathbf{x}_i) \neq a_i\} \tag{4.14}$$

Following (4.14), the first term in (4.13) will be $\mathbf{w}^{*\top} \sum_{t=1}^{T} \eta_t \mathbf{d}_t \geq 0$ and thus, can be ignored. Moreover, by the definition of γ, the second term is $\sum_{t=1}^{T} -\mathbf{w}_t^{\top} \eta_t \mathbf{d}_t = \sum_{t=1}^{T} \eta_t \gamma_t$. Putting all this together, ignoring the last term in (4.13) and replacing $||\mathbf{d}_t||_*^2$ with its upper bound L, yields

$$-B_{\mathcal{R}}(\mathbf{w}^*, \mathbf{w}_1) \leq L \sum_{t=1}^{T} \frac{1}{2} \eta_t^2 - \sum_{t=1}^{T} \eta_t \gamma_t \tag{4.15}$$

Replacing the left side with $-B_{\mathcal{R}} = -\frac{1}{2}||\mathbf{w}^* - \mathbf{w}_1||^2 = \frac{\epsilon - 1}{2N\epsilon}$ for the case of quadratic $\mathcal{R}$, and with $-B_{\mathcal{R}} = \log(\epsilon)$ when $\mathcal{R}$ is a negative entropy function, taking the derivative with respect to η_t and equating it to zero (which yields $\eta_t = \frac{\gamma_t}{L}$), the error bounds in (4.8) and (4.9) are achieved. Note that in the case of $\mathcal{R}$ being the negative entropy function, Algorithm 4.1 degenerates into ADABoost with a different choice of η_t.

Before continuing our discussion, it is important to mention that the cornerstone concept of the proof is the choice of $\mathbf{w}^*$. For instance, a different choice of $\mathbf{w}^*$ results in the following maximum margin theorem

Theorem 4.7. Maximum margin property of MABoost *Setting $\eta_t = \frac{\gamma_t}{L\sqrt{t}}$, MABoost outputs a hypothesis of margin at least $\gamma_{min} - \nu$, where ν is a desired accuracy level and tends to zero in $O(\frac{\log T}{\sqrt{T}})$ rounds of boosting.*

See Appendix A.2 for the proof.

Observations: Two observations follow immediately from the proof of Theorem 4.6. First, the requirement of using Lemma 4.2 is $\mathbf{w}^* \in \mathcal{S}$, so in the case of projecting onto a smaller convex set $\mathcal{S}_k \subseteq \mathcal{S}$, as long as $\mathbf{w}^* \in \mathcal{S}_k$ holds, the proof is intact. Second, only the relaxed version of Lemma 1 is required in the proof (to obtain inequality (4.10c)). Hence, if there is an approximate projection operator $\hat{\Pi}_{\mathcal{S}}$ that satisfies the inequality $B_{\mathcal{R}}(\mathbf{w}^*, \mathbf{z}_{t+1}) \geq B_{\mathcal{R}}\big(\mathbf{w}^*, \hat{\Pi}_{\mathcal{S}}(\mathbf{z}_{t+1})\big)$, it can be substituted for the exact projection operator $\Pi_{\mathcal{S}}$ and the active update version of the algorithm still works. A practical approximate projection operator of this type can be obtained through a double-projection strategy (see Appendix A.3 for the proof).

Lemma 4.8. *Consider the convex sets $\mathcal{K}$ and $\mathcal{S}$, where $\mathcal{S} \subseteq \mathcal{K}$. Then for any $\mathbf{x} \in \mathcal{S}$ and $\mathbf{y} \in \mathbb{R}^N$, $\hat{\Pi}_{\mathcal{S}}(\mathbf{y}) = \Pi_{\mathcal{S}}\Big(\Pi_{\mathcal{K}}(\mathbf{y})\Big)$ is an approximate projection that satisfies $B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) \geq B_{\mathcal{R}}\big(\mathbf{x}, \hat{\Pi}_{\mathcal{S}}(\mathbf{y})\big)$.*

The above observations are employed to generalize Algorithm 4.1. However, it is important to emphasize that the approximate Bregman projection is only valid for the active update version of MABoost due to the fact that the relaxed version of Lemma 4.2 used in the proof of Theorem 4.6 is only valid for the active update version.

4.4.1 Sparse Boosting

Let $\mathcal{R}(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||_2^2$. Since in this case the projection onto the probability simplex is in fact an ℓ_1-constrained optimization problem, it is plausible that some of the weights are zero (sparse distribution), which is already a useful observation. To promote the sparsity of the weight vector, it is desirable to directly regularize the projection with the ℓ_1 norm, i.e., adding $||\mathbf{w}||_1$ to the objective function in the projection step. It is, however, not possible in MABoost, since $||\mathbf{w}||_1$ is trivially constant on the probability simplex. Therefore, the projection step is split into two consecutive steps. The first projection is onto $\mathcal{R}_+^N = \{\mathbf{y} \mid 0 \leq y_i\}$.

Surprisingly, projection onto $\mathcal{R}_+^N$ implicitly regularizes the weights of the correctly classified samples with a weighted ℓ_1 norm term. This point is clearly shown in the proof of Theorem 4.9 presented in Appendix A.5. To further enhance sparsity, we may introduce an explicit ℓ_1 norm regularization term into the projection step with a regularization factor denoted by $\alpha_t \eta_t$. The solution of the projection step is then normalized to get a feasible point on the

probability simplex. This algorithm is listed in Algorithm 4.2. $\alpha_t\eta_t$ is the regularization factor of the explicit ℓ_1 norm at round t. Note that the dominant regularization factor is $\eta_t d_t^i$ which only pushes the weights of the correctly classified samples to zero, i.e., when $d_t^i < 0$. This can become evident by substituting the update step in the projection step for $\mathbf{z}_{t+1}$.

Algorithm 4.2: SparseBoost

Input: $\mathcal{R}(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||_2^2, \mathcal{R}_+^N = \{\mathbf{y} \mid 0 \leq y_i\}, \mathbf{y}_1 = [\frac{1}{N}, \ldots, \frac{1}{N}]^\top$

For $t = 1, \ldots, T$ **do**

(a) Train classifier with $\mathbf{w}_t$ and get h_t.

(b) Set $(\eta_t = \frac{\gamma_t||y_t||_1}{N}, \alpha_t = 0)$ or $(\eta_t = \frac{\gamma_t||y_t||_1}{2N}, \alpha_t = \frac{1}{2}\gamma_t||y_t||_1)$.

(c) $\mathbf{z}_{t+1} = \mathbf{y}_t + \eta_t \mathbf{d}_t$.

(d) $\mathbf{y}_{t+1} = \arg\min_{\mathbf{y} \in \mathcal{R}_+^N} \frac{1}{2}||\mathbf{y} - \mathbf{z}_{t+1}||^2 + \alpha_t\eta_t||\mathbf{y}||_1$.

(e) $\mathbf{w}_{t+1} = \frac{\mathbf{y}_{t+1}}{\sum_{i=1}^N y_t^i}$.

End

Output: The final hypothesis $f(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^T \eta_t h_t(\mathbf{x})\right)$.

For simplicity, two cases are considered: when $\alpha_t = \min(1, \frac{1}{2}\gamma_t||\mathbf{y}_t||_1)$ and when $\alpha_t = 0$. The training error bound is given by the following theorem (proof in Appendix A.5):

Theorem 4.9. *Suppose that SparseBoost generates weak hypotheses $h_1, \ldots, h_T$ whose edges are $\gamma_1, \ldots, \gamma_T$. Then the error ϵ of the combined hypothesis f on the training set is bounded as follows:*

$$\epsilon \leq \frac{1}{1 + c\sum_{t=1}^T \gamma_t^2 ||\mathbf{y}_t||_1^2} \tag{4.16}$$

Note that this bound holds for any choice of $\alpha \in \big[0, \min(1, \gamma_t||\mathbf{y}_t||_1)\big)$. Particularly, in our two cases constant c is 1 for $\alpha_t = 0$, and $\frac{1}{4}$ when $\alpha_t = \min(1, \frac{1}{2}\gamma_t||\mathbf{y}_t||_1)$.

For $\alpha_t = 0$, the ℓ_1 norm of the weights $||\mathbf{y}_t||_1$ can be bounded away from zero by $\frac{1}{N}$ (see Appendix A.5). Thus, the error ϵ tends to zero by $O(\frac{N^2}{\gamma^2 T})$. That is, in this case Sparseboost is a provable boosting algorithm. However, for $\alpha_t \neq 0$, the ℓ_1 norm $||\mathbf{y}_t||_1$ may rapidly go to zero and consequently the upper bound of the training error in (4.16) does not vanish as T increases.

In this case, it is not possible to conclude that the algorithm is in fact a boosting algorithm[6]. It is noteworthy that SparseBoost can be seen as a variant of the COMID algorithm in [DSST10].

4.4.2 Smooth Boosting

Let $k > 0$ be a smoothness parameter. A distribution $\mathbf{w}$ is smooth with respect to a given distribution $\mathbf{D}$ if $w_i \leq kD_i$ for all $1 \leq i \leq N$. Here, the smoothness with respect to to the uniform distribution, i.e., $D_i = \frac{1}{N}$, is considered. Then, given a desired smoothness parameter k, a boosting algorithm is required that only constructs distributions $\mathbf{w}$ such that $w_i \leq \frac{k}{N}$, while guaranteeing to output a $(1-\frac{1}{k})$-accurate hypothesis. To this end, it is only required to replace the probability simplex $\mathcal{S}$ with $\mathcal{S}_k = \{\mathbf{w} | \sum_{i=1}^{N} w_i = 1, 0 \leq w_i \leq \frac{k}{N}\}$ in MABoost to obtain a smooth distribution boosting algorithm, called smooth-MABoost. That is, the update rule is: $\mathbf{w}_{t+1} = \operatorname*{argmin}_{\mathbf{w} \in \mathcal{S}_k} B_{\mathcal{R}}(\mathbf{w}, \mathbf{z}_{t+1})$.

Note that the proof of Theorem 4.6 holds for smooth-MABoost, as well. As long as $\epsilon \geq \frac{1}{k}$, the error distribution $\mathbf{w}^*$ ($w_i^* = \frac{1}{N\epsilon}$ if $f(\mathbf{x}_i) \neq a_i$, and 0 otherwise) is in $\mathcal{S}_k$ because $\frac{1}{N\epsilon} \leq \frac{k}{N}$. Thus, based on the first observation, the error bounds achieved in Theorem 4.6 hold for $\epsilon \geq \frac{1}{k}$. In particular, $\epsilon = \frac{1}{k}$ is reached after a finite number of iterations. This projection problem has already appeared in the literature. An entropic projection algorithm ($\mathcal{R}$ is negative entropy), for instance, was proposed in [SS08]. Using the negative entropy and the suggested projection algorithm in [SS08] results in a fast smooth boosting algorithm with the following convergence rate.

Theorem 4.10. *Given $\mathcal{R}(\mathbf{w}) = \sum_{i=1}^{N} w_i \log w_i$ and a desired ϵ, smooth-MABoost finds a $(1-\epsilon)$-accurate hypothesis in $O(\log(\frac{1}{\epsilon})/\gamma^2)$ of iterations.*

4.4.3 MABoost for Combining Datasets (CD-MABoost)

Let's assume we have two sets of data. A primary dataset $\mathcal{A}$ and a secondary dataset $\mathcal{B}$. The goal is to train a classifier that achieves $(1-\epsilon)$ accuracy on $\mathcal{A}$ while limiting the error on dataset $\mathcal{B}$ to $\epsilon_{\mathcal{B}} \leq \frac{1}{k}$. This scenario has many potential applications including (I) transfer learning [DYXY07] and (II) weighted combination of datasets based on their noise level or emphasizing on a particular region of a sample space as a problem requirement (e.g., a

[6]Nevertheless, for some choices of $\alpha_t \neq 0$ such as $\alpha_t \propto \frac{1}{t^2}$, the boosting property of the algorithm is still provable.

medical diagnostic test that should make as few wrong diagnoses as possible when the sample is a pregnant woman). To address this problem, it is only required to replace $\mathcal{S}$ in MABoost with $\mathcal{S}_c = \{\mathbf{w} | \sum_{i=1}^{N} w_i = 1, 0 \leq w_i \ \forall i \in \mathcal{A} \ \wedge \ 0 \leq w_i \leq \frac{k}{N} \ \forall i \in \mathcal{B}\}$ where $i \in \mathcal{A}$ shorthands the indices of samples in $\mathcal{A}$. By generating smooth distributions on $\mathcal{B}$, this algorithm limits the weight of the secondary dataset, which intuitively results in limiting its effect on the final hypothesis. The proof of its boosting property is quite similar to Theorem 4.6 (see Appendix A.4).

4.4.4 Lazy Update Boosting

In this section, the proof for the lazy update version of MABoost in Theorem 4.6 is presented. Moreover, it is shown that MADABoost [DW00] can be presented as a variant of the lazy update MABoost. This gives a simple proof for MADABoost without making the assumption that the edge sequence is monotonically decreasing (as in [DW00]).

Proof: Assume $\mathbf{w}^* = [w_1^*, \dots, w_N^*]^\top$ is a distribution vector where $w_i^* = \frac{1}{N\epsilon}$ if $f(\mathbf{x}_i) \neq a_i$, and 0 otherwise. Then,

$$
\begin{aligned}
(\mathbf{w}^* - \mathbf{w}_t)^\top \eta_t \mathbf{d}_t =& (\mathbf{w}_{t+1} - \mathbf{w}_t)^\top \big(\nabla \mathcal{R}(\mathbf{z}_{t+1}) - \nabla \mathcal{R}(\mathbf{z}_t)\big) \\
&+ (\mathbf{z}_{t+1} - \mathbf{w}_{t+1})^\top \big(\nabla \mathcal{R}(\mathbf{z}_{t+1}) - \nabla \mathcal{R}(\mathbf{z}_t)\big) \\
&+ (\mathbf{w}^* - \mathbf{z}_{t+1})^\top \big(\nabla \mathcal{R}(\mathbf{z}_{t+1}) - \nabla \mathcal{R}(\mathbf{z}_t)\big) \\
\leq& \frac{1}{2} ||\mathbf{w}_{t+1} - \mathbf{w}_t||^2 + \frac{1}{2} \eta_t^2 ||\mathbf{d}_t||_*^2 + B_{\mathcal{R}}(\mathbf{w}_{t+1}, \mathbf{z}_{t+1}) \\
&- B_{\mathcal{R}}(\mathbf{w}_{t+1}, \mathbf{z}_t) + B_{\mathcal{R}}(\mathbf{z}_{t+1}, \mathbf{z}_t) \\
&- B_{\mathcal{R}}(\mathbf{w}^*, \mathbf{z}_{t+1}) + B_{\mathcal{R}}(\mathbf{w}^*, \mathbf{z}_t) - B_{\mathcal{R}}(\mathbf{z}_{t+1}, \mathbf{z}_t) \\
\leq& \frac{1}{2} ||\mathbf{w}_{t+1} - \mathbf{w}_t||^2 + \frac{1}{2} \eta_t^2 ||\mathbf{d}_t||_*^2 - B_{\mathcal{R}}(\mathbf{w}_{t+1}, \mathbf{w}_t) \\
&+ B_{\mathcal{R}}(\mathbf{w}_{t+1}, \mathbf{z}_{t+1}) - B_{\mathcal{R}}(\mathbf{w}_t, \mathbf{z}_t) \\
&- B_{\mathcal{R}}(\mathbf{w}^*, \mathbf{z}_{t+1}) + B_{\mathcal{R}}(\mathbf{w}^*, \mathbf{z}_t) \qquad (4.17)
\end{aligned}
$$

where the first inequality follows from applying Lemma 4.5 to the first term and Lemma 4.3 to the rest of the terms and the second inequality is the result of applying the exact version of Lemma 4.2 to $B_{\mathcal{R}}(\mathbf{w}_{t+1}, \mathbf{z}_t)$. Moreover, since $B_{\mathcal{R}}(\mathbf{w}_{t+1}, \mathbf{w}_t) - \frac{1}{2} ||\mathbf{w}_{t+1} - \mathbf{w}_t||^2 \geq 0$, it can be ignored in (4.17).

Summing up the inequality (4.17) from $t = 1$ to T, yields

$$-B_{\mathcal{R}}(\mathbf{w}^*, \mathbf{z}_1) \leq L \sum_{t=1}^{T} \frac{1}{2}\eta_t^2 - \sum_{t=1}^{T} \eta_t \gamma_t \tag{4.18}$$

where the facts that $\mathbf{w}^{*\top} \sum_{t=1}^{T} \eta_t \mathbf{d}_t \geq 0$ and $\sum_{t=1}^{T} -\mathbf{w}_t^\top \eta_t \mathbf{d}_t = \sum_{t=1}^{T} \eta_t \gamma_t$ are used. Inequality 4.18 is exactly the same as (4.15), and replacing $-B_{\mathcal{R}}$ with $\frac{\epsilon-1}{N\epsilon}$ or $\log(\epsilon)$ yields the same error bounds in Theorem 4.6. Note that, since the exact version of Lemma 4.2 is required to obtain (4.17), this proof does not reveal whether MABoost can be generalized to employ the double-projection strategy. In some particular cases, however, it maybe possible to show that a double-projection variant of MABoost is still a provable boosting algorithm.

In the following, we briefly show that MADABoost can be seen as a double-projection lazy MABoost .

Algorithm 4.3: Variant of MADABoost

Let $\mathcal{R}(\mathbf{w})$ be the negative entropy and $\mathcal{K}$ a unit hypercube; Set $\mathbf{z}_1 = [1, \ldots, 1]^\top$;

At $t = 1, \ldots, T$, train h_t with $\mathbf{w}_t$, set $f_t(\mathbf{x}) = \text{sign}\left(\sum_{t'=1}^{t} \eta_{t'} h_{t'}(\mathbf{x}) \right)$ and calculate $\epsilon_t = \frac{\sum_{i=1}^{N} \frac{1}{2}|f_t(\mathbf{x}_i) - a_i|}{N}$, set $\eta_t = \epsilon_t \gamma_t$ and update

(a) $\nabla\mathcal{R}(\mathbf{z}_{t+1}) = \nabla\mathcal{R}(\mathbf{z}_t) + \eta_t \mathbf{d}_t \quad \rightarrow z_{t+1}^i = z_t^i e^{\eta_t d_t^i}$

(b) $\mathbf{y}_{t+1} = \arg\min_{\mathbf{y} \in \mathcal{K}} B_{\mathcal{R}}(\mathbf{y}, \mathbf{z}_{t+1}) \quad \rightarrow y_{t+1}^i = \min(1, z_{t+1}^i)$

(c) $\mathbf{w}_{t+1} = \arg\min_{\mathbf{w} \in \mathcal{S}} B_{\mathcal{R}}(\mathbf{w}, \mathbf{y}_{t+1}) \quad \rightarrow w_{t+1}^i = \frac{y_{t+1}^i}{||\mathbf{y}_{t+1}||_1}$

Output the final hypothesis $f(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^{T} \eta_t h_t(\mathbf{x}) \right)$.

Algorithm 4.3 is essentially MADABoost, only with a different choice of η_t. It is well-known that projection of a weight vector via entropy function onto the probability simplex results in a vector which is the ℓ_1 normalized of the original vector. This explains the normalization in step (b). The entropy projection onto the unit hypercube (i.e., step (c)), however, maybe less known and thus, its proof is given in Appendix A.6. The following theorem gives the

convergence rate of the MADABoost algorithm (see Appendix A.7 for the proof).

Theorem 4.11. *Algorithm 4.3 yields a* $(1-\epsilon)$*-accurate hypothesis after at most* $T = O(\frac{1}{\epsilon^2\gamma^2})$.

This is an important result since it shows that MADABoost seems, at least in theory, to be slower than what we hoped, namely $O(\frac{1}{\epsilon\gamma^2})$.

4.5 Multiclass Generalization

The convergence proof of the MABoost framework (and the algorithms derived from it) is based on the well-understood weak-learning assumption stating that the weak classifiers predict better than random on any distribution over the training set. Logically, in order to generalize the MABoost framework to multiclass setting, the weak-learning assumption need to be generalized. However, as it has already been shown (see [SS98],[ZZRH09] and [MS13]), it is not trivial. While there is a unique definition for weak-learnability in binary setting, infinite number of weak-learning assumptions can be taken in multiclass classification. For instance, the straightforward extension of the binary weak-learning condition requires that each classifier only predicts better than random guessing, that is, given K classes the performance of a weak classifier should only be better than $1/K$. The SAMME algorithm in [ZZRH09], for instance, adopts this assumption. However, as the number of classes increases, the random guessing baseline approaches zero and consequently the weak-learning requirement vanishes. It is hence not surprising that this condition turned out to be too weak for being boostable, i.e., no boosting algorithm with this assumption can drive the training error to zero [MS13]. Another possible weak-learning assumption is to require the classifiers to achieve more than 50% accuracy on any distribution over the training data. This assumption increasingly becomes difficult to satisfy as the number of classes increases. Since most of simple classifiers (such as decision stumps) fail to meet this requirement, by adopting this assumption we need to employ more complex weak learners which may increase the overfitting tendency. In their groundbreaking paper [MS13], Mukherjee and Schapire argued that the optimal condition is the weakest condition that still guarantees the boostability. They derived the necessary and sufficient weak-learning condition for boostability, called minimal weak-learning condition,

and more importantly they presented an inequality constraint which can be used to check whether a given weak-learning assumption is equivalent to the minimal weak-learning condition.

Throughout the rest of Section 4.5, the MABoost framework is generalized to the multiclass setting and by using the results derived in [MS13] it is shown that it adopts the minimal weak-learning condition.

4.5.1 Preliminaries for Multiclass Setting

Let $\{(\mathbf{x}_i, a_i)\}, 1 \leq i \leq N$, be N training samples, where $\mathbf{x}_i \in \mathcal{X}$ are feature vectors and $a_i \in \{0, \ldots, K-1\}$ are class labels. Assume $h \in \mathcal{H}$ is a discrete-valued function mapping $\mathcal{X}$ into $\{0, \ldots, K-1\}$. For the sake of simplicity, it is assumed that all samples belong to class 0. Nevertheless, we may refer to the label of sample i by a_i or by 0 depending on the context. This assumption highly simplifies our presentation.

Assume an $N \times K$ weight matrix whose $(i, j)^{\text{th}}$ element represents the non-negative weight of wrongly classifying sample i to be in class $j + 1$ and the sum of each row is zero, i.e., $(i, 1)$ element (since all labels are 0) is a negative weight equal to the minus of the sum of non-negative weights. Due to this zero-sum constraint, the first column does not contain any additional information and can be discarded. Our weight matrix $\mathbf{W}$ is then an $N \times K-1$ matrix whose $(i, j)^{\text{th}}$ elements $W_{i,j}$ are non-negative costs of wrongly predicting the label of the i^{th} sample to be j.

Analogously to the binary setting, a definition for the loss matrix (vector in the binary setting) is required. In the binary setting, the i^{th} element of the loss vector $\mathbf{d}$ was -1, if the i^{th} sample was classified correctly, and 1 otherwise. In the multiclass case, a loss matrix $\mathbf{D}$ is defined as an $N \times K-1$ matrix where all the elements of the i^{th} row are -1, if the hypothesis h classifies the i^{th} sample correctly. However, if h makes a mistake and assigns the i^{th} sample to the j^{th} class, then $D_{i,j}$ is 1 and the rest of the elements of the i^{th} row are zero. That is, given a weak classifier h, the elements of the loss matrix $\mathbf{D}$ are:

$$D_{i,j} = \begin{cases} -1, & \text{if } h(\mathbf{x}_i) = 0 \\ 0, & \text{if } h(\mathbf{x}_i) = l \text{ and } j \neq l \\ 1, & \text{if } h(\mathbf{x}_i) = j \end{cases} \tag{4.19}$$

With these definitions at hand, the weak-learning assumption can be defined as:

Definition 4.12. *A weak hypothesis space $\mathcal{H}$ satisfies the weak-learning condition if for any element-wise nonnegative $\mathbf{W}$, there is a hypothesis $h \in \mathcal{H}$ with loss matrix $\mathbf{D}$ as defined in (4.19) that is*

$$-\mathbf{D} \bullet \mathbf{W} \geq \gamma \qquad \exists\, \gamma \geq 0. \tag{4.20}$$

where the inner product of two matrices $\mathbf{A}$ and $\mathbf{B}$ is defined as the trace of their products $\text{Tr}\{\mathbf{A}\mathbf{B}^\top\}$ and denoted by $\mathbf{A}\bullet\mathbf{B}$. Without loss of generality, in this definition it is assumed that $\mathbf{W}$ is normalized so that the sum of its elements is 1. It is shown in Appendix A.8 that this definition for weak-learning condition is equivalent to that of ADABoost.MR [SS98]. As shown in [MS13], ADABoost.MR satisfies the minimal weak-learning condition. Thus, any hypothesis space $\mathcal{H}$ that satisfies the weak-learning condition in Definition 3 is in fact a boostable hypothesis space.

4.5.2 Multiclass MABoost (Mu-MABoost)

Assume $\mathbf{X}$ is a matrix. A matrix function $\mathcal{R}(\mathbf{X})$ is defined to be equal to $\mathcal{R}(\mathbf{x})$, where $\mathbf{x}$ is a super-vector constructed by concatenating all the columns of $\mathbf{X}$. Let $\mathcal{R}(\mathbf{X})$ be a 1-strongly convex function with respect to a norm $||.||$ and denote its associated Bregman divergence with $B_{\mathcal{R}}$. Moreover, let the dual norm of a loss matrix $\mathbf{D}_t$ be defined as the dual norm of its super-vector $\mathbf{d}$ (constructed by concatenating its columns) and assume it is bounded from above, i.e., $||\mathbf{D}_t||_*=||\mathbf{d}_t||_* \leq L$. It is easy to verify that for $\mathbf{D}_t$ in (4.19), $L=1$ when $||.||_* = \ell_\infty$ and $L= N(K-1)$ when $||.||_* = \ell_2$. Algorithm 4.4 is then the generalization of the MABoost framework to the multiclass setting. In the output of Algorithm 4.4, $\mathbf{1}(.)$ is an indicator function which returns 1 if its argument holds, and zero otherwise.

As in the binary MABoost framework, it can be shown that if the weak classifiers are selected from a boostable space, the training error of the Mu-MABoost algorithms approaches zero in finite number of rounds.

Theorem 4.13. *Suppose that Mu-MABoost generates weak hypotheses $h_1, \ldots, h_T$ whose edges are $\gamma_1, \ldots, \gamma_T$. Then the error ϵ of the combined*

hypothesis f on the training set is bounded as:

$$\mathcal{R}(\mathbf{W}) = \frac{1}{2}||\mathbf{W}||_2^2 : \qquad \epsilon \leq \frac{1}{1+\sum_{t=1}^{T}\gamma_t^2}$$

$$\mathcal{R}(\mathbf{W}) = \sum_{i=1}^{N(K-1)} w_i \log w_i : \qquad \epsilon \leq e^{-\sum_{t=1}^{T}\frac{1}{2}\gamma_t^2}$$

$\mathcal{R}(\mathbf{W})$ is equal to $\mathcal{R}(\mathbf{w})$ with $\mathbf{w}$ being the super-vector representing $\mathbf{W}$. The proof of Theorem 4.13 is in the spirit of the proof of Theorem 4.6. The only difference is that instead of an error vector, an $N \times K-1$ error matrix $\mathbf{W}^*$ is used where the elements of the i^{th} row are $\frac{1}{N(K-1)}$ if f classifies $\mathbf{x}_i$ wrongly. Replacing this error matrix in the proof of Theorem 4.6 and recalling the fact that matrix function $\mathcal{R}(\mathbf{X})$ is defined to be $\mathcal{R}(\mathbf{x})$ (with $\mathbf{x}$ being a super-vector constructed by concatenating the columns) yields the proof of Theorem 4.13.

Algorithm 4.4: Multiclass MABoost (Mu-MABoost)

Input: $\mathcal{R}(\mathbf{X})$ 1-strongly convex function,
$\mathbf{W}_1$ and $\mathbf{Z}_1$ with elements $W_{i,j}^1 = Z_{i,j}^1 = \frac{1}{N(K-1)}$

For $t = 1, \ldots, T$ **do**

(a) Train classifier with $\mathbf{W}_t$ and get h_t, set $\mathbf{D}_t$ elements as in (4.19) and $\gamma_t = -\mathbf{W}_t \bullet \mathbf{D}_t$.

(b) Set $\eta_t = \frac{\gamma_t}{L}$

(c) Update weights: $\nabla\mathcal{R}(\mathbf{Z}_{t+1}) = \nabla\mathcal{R}(\mathbf{Z}_t) + \eta_t\mathbf{D}_t$ (lazy update)
$\nabla\mathcal{R}(\mathbf{Z}_{t+1}) = \nabla\mathcal{R}(\mathbf{W}_t) + \eta_t\mathbf{D}_t$ (active update)

(d) Project onto $\mathcal{S}$: $\mathbf{W}_{t+1} = \underset{\mathbf{W}\in\mathcal{S}}{\operatorname{argmin}} B_{\mathcal{R}}(\mathbf{W}, \mathbf{Z}_{t+1})$

End

Output: Final hypothesis $f(\mathbf{x})$: $H(\mathbf{x}, l) = \sum_{t=1}^{T}\eta_t\mathbf{1}\big(h_t(\mathbf{x}) == l\big)$
$f(\mathbf{x}) = \underset{l}{\operatorname{argmin}}\, H(\mathbf{x}, l)$

It is noteworthy that using KL-divergence as the Bregman divergence in Algorithm 4.4 gives a version of ADABoost.MM introduced in [MS13] with slightly different values for η.

4.6 Classification Experiments with Boosting

In this section, the experiments that were run to evaluate the boosting algorithms presented in this work are described. For evaluation of binary and multiclass algorithms, 13 datasets from the UCI repository were used. They contain all combinations of real and discrete features, are drawn from various real-world problems and have been regularly used in previous works as learning benchmark problems. Tables 4.1 and 4.2 list the binary and multiclass datasets and their descriptions, respectively, and provide some properties of these datasets.

Dataset Name	Features	Training samples	Test samples
Breast cancer	9	549	150
German-credit	20	700	300
Sonar	60	158	50
Gisette*	100	40×100	2000
Pima-diabetes	8	500	168
House-votes-84	16	335	100
Thyroid-disease	25	1500	500

Table 4.1: Binary datasets description

In all experiments with binary datasets, the experiments were run for 20 times over randomly selected training and test sets with the sample numbers specified in Table 4.1 and the results averaged. The only exception for this test procedure is the Gisette dataset whose original feature vector dimension is 5000. For this dataset, we first selected 100 features with the highest mutual

Dataset Name	Features	Training samples	Test samples	Classes
Abalone	8	3177	1000	28
Connect-4	42	57577	10000	3
Car	6	1228	500	4
Forest	54	20000	10000	7
Letter	16	15000	5000	26
Poker	10	15010	10000	10

Table 4.2: Multiclass datasets description.

information values with the class labels and then set aside 2000 samples as the test set. The remaining 4000 samples were then divided to 40 training sets each containing 100 samples. Each algorithm was then run on these training sets and its test errors were averaged.

For multiclass problems, however, most sets have more than 10000 training samples and 5000 test samples giving sufficient confidence on the test results. Thus, the experiments were run only once. Moreover, the Forest dataset is too large and contains more than 500000 samples. Thus, only 20000 of its samples were used for training and 10000 for test.

4.6.1 Binary MABoost Experiments

In the first set of experiments, the performance of the algorithms in the presence of mislabeling noise was evaluated. Moreover, two different weak learners were utilized in MABoost: Conventional decision trees and a special implementation of decision trees. In this implementation, at each round of training a small set of features (precisely speaking $\sqrt{M}$ features where M is the total number of features) were selected. A random cost was assigned then to each of the chosen features. These costs were uniformly drawn from $\mathcal{U}(1, 4)$ and are in fact scaling coefficients to be applied when considering splits, so the improvement on splitting on a feature is divided by its cost in deciding which split to choose [TA$^+$97]. Given these costs, a decision tree was then grown with the selected features. This particular implementation of weak

Data set	MABoost	MA-Forest	Real-ADA	RF
Breast cancer	3.96	3.21	3.87	**3.08**
German-credit	**23.24**	24.82	23.36	23.89
Votes-84	3.99	3.90	3.96	**3.71**
Pima-diabetes	23.55	24.46	23.59	**23.53**
Thyroid-disease	**2.53**	3.12	2.60	2.83
Sonar	16.23	16.54	**13.88**	17.11
Gisette	11.95	**11.04**	11.73	11.78

Table 4.3: The test errors in percentage of binary classification with no mislabeling noise for four algorithms. Each algorithm grows 500 decision trees with a maximum size of 63. MA-Forest, Real-ADA and RF stand for MABoost-Forest, real-ADABoost and Random forest, respectively.

learner can be found in MABoost R package available on CRAN [Nag14]. We empirically found that this random-forest-type weak learner usually reduces the generalization error. Each of the algorithms were growing and combining 500 trees and the size of the trees was limited to 63.

Data set	MABoost	MA-Forest	Real-ADA	RF
Breast cancer	4.48	**3.89**	3.94	4.60
German-credit	25.80	25.81	**25.40**	25.83
Votes-84	5.41	**4.83**	5.62	5.76
Pima-diabetes	**25.16**	25.22	25.36	25.78
Thyroid-disease	5.07	**4.8**	5.11	5.01
Sonar	21.27	**19.87**	21.67	20.00
Gisette	15.85	14.20	16.98	**14.15**

Table 4.4: The test errors of binary classification with 15% mislabeling noise in the training data. Each algorithm grows 500 decision trees of size limited to 63. A bold value at each row represents the lowest test error for that dataset.

Tables 4.3 and 4.4 report the performance of MABoost, random-forest type MABoost which is called MABoost-Forest, real ADABoost and random forest. In the first scenario (reported in Table 4.3), the mislabeling noise was zero while in the second scenario 15% of the training samples had wrong labels. As can be seen, in the absence of noise, random forest works better than boosting methods for three out of seven datasets. Due to the inherent robustness of random-subspace-selection based ensemble methods such as random forest against mislabeling noise, it is expected that Random forest wins in the noisy scenario as well. Surprisingly, MABoost-Forest yields significantly better performance than other methods in the presence of 15% mislabeling noise. MABoost-Forest gives higher accuracy for four out of seven datasets and for those three that it does not win, its accuracy is close to that of the winner. It is perhaps due to the fact that MABoost-Forest inherits the robustness of Random forest while taking advantage of bias correction power of boosting methods.

4.6.2 Experiment with SparseBoost

Reducing the memory complexity is the main advantage of the SparseBoost algorithm. SparseBoost reduces the memory complexity by utilizing only a

percentage of training samples at each round of boosting. In our experiments, the efficiency of the SparseBoost algorithm in the sense of memory complexity reduction is evaluated. The second column of Table 4.5 reports the average sparsity ratio of the weight vectors. The sparsity ratio is defined as the number of zero weights to the total number of weights and the average was taken over the sparsity ratios of the 200 weight vectors generated during the boosting process. This number is reported in percentage. As can be

Data set	**Sparsity ratio%**	**Training err%**	**Test err%**
Breast cancer	67.03	0	4.13
German-credit	25.62	0	25.00
Votes-84	49.35	0.59	4
Pima-diabetes	25.22	25.36	25.78
Thyroid-disease	40.30	0.03	4.73
Sonar	42.57	0	24.30
Gisette	47.84	0	13.70

Table 4.5: The tables shows the result of the experiments with SparseBoost algorithm. All values are in percentage.

seen in Table 4.5, for most of the datasets, at each round of boosting over 40% of the samples were not used. For the Breast cancer dataset, the sparsity ratio is particularly interesting: only 33% of training samples per round were necessary to construct a perfect classifier in the sense of having 100% accuracy on the training set. However, this complexity reduction comes at the expense of higher generalization error. Comparing the accuracies reported in Table 4.3 and Table 4.5 reveals that the SparseBoost algorithm consistently achieves lower classification accuracy than MABoost, mostly due to the fact that at each round of boosting, SparseBoost constructs an *easier* dataset by excluding some of the samples from the training data. Even though the *easy* samples might have small weights in the case of using other weighting mechanisms (such as ADABoost), they still affect the training procedure by acting as a regularizer by preventing the algorithm to perfectly fit the hard samples. Hence, completely excluding them from the dataset increases the chance of overfitting which consequently reduces the generalization power.

4.6.3 Experiment with CD-MABoost

Assume for a particular classification task, several datasets with different qualities are given. The quality of a dataset depends on several factors such as the level of the feature noise and frequency of labeling errors. When the quality of the datasets are approximately known, we may incorporate this information into our learning procedure by restricting the weights of the low-quality samples. As discussed in Section 4.4.3, CD-MABoost takes this approach to combine datasets. In CD-MABoost, the distribution over the good-quality samples is unconstrained while the weights of the low-quality samples are limited to be less than a given smoothness threshold $1/k$.

An experiment was run to show the effectiveness of this method for combining different quality datasets. To this end, an artificially generated dataset suggested by Long and Servedio in [LS10] was utilized. This dataset (which is called L-S dataset in this chapter) contains 4000 training samples (no teset set is needed in this experiment). Each sample (x, y) in this dataset is generated as follows. First the label y is chosen randomly from $\{-1, 1\}$. There are 21 features $x_1, \ldots, x_{21}$ that take values in $\{-1, 1\}$. The features of 1000 of samples (called large margin samples) are set as: $x_1 = \ldots = x_{21} = y$. Another 1000 samples (called puller samples) are set to: $x_1 = \ldots = x_{11} = y$ and $x_{12} = \ldots = x_{21} = -y$. The rest of the samples (i.e. 2000 samples) which are called penalizers are chosen in there stages: (I) The values of a random subset of five of the first eleven features $x_1, \ldots, x_{11}$ are set equal to y, (II) the values of a random subset of six of the last ten features $x_{12}, \ldots, x_{21}$ are set equal to y, and (III) the remaining ten features are set to $-y$.

This data, as it is, can be fully learned by an ensemble of decision stumps. However, as has been discussed in [LS10], by only adding 10% mislabeling noise to this dataset (by randomly flipping the labels) no boosting algorithm with a convex loss function can learn the underlying hypothesis. In fact, with only 10% mislabeling noise in the data, neither of the commonly used boosting methods such as MADABoost, ADABoost and LogitBoost algorithms can drive the training error below 27%. By means of this dataset it will be shown below that our CD-MABoost algorithm (see Section 4.4.3) does not have this limitation.

Assume two L-S datasets are given. One of these datasets is corrupted by 20% mislabeling noise while the other is clean, i.e., if these two datasets are combined, we have 10% mislabeling noise in the final dataset. While, as extensively discussed in [LS10], it is impossible to learn this dataset with a

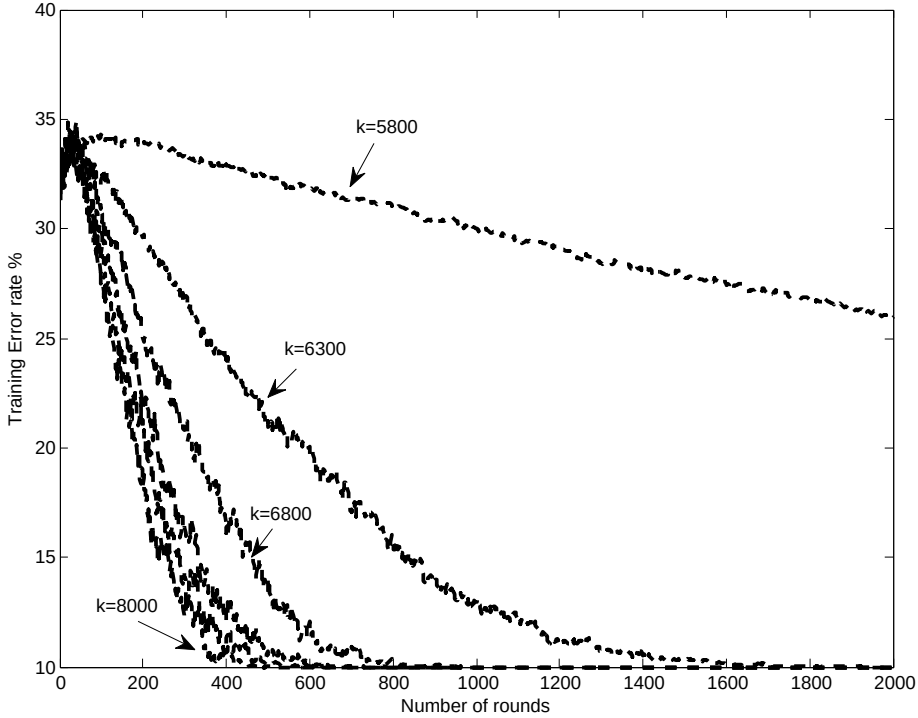

Figure 4.1: The training error of CD-MABoost over Long-Servedio dataset with 10% error for different values of k (the weights of the mislabeled samples are restricted to be smaller than $1/k$).

convex loss based boosting method, it is shown here that by restricting the weights of the corrupted dataset (to be less than $1/k$), this dataset can be effectively learned by CD-MABoost. In Figure 4.1 the training error of this boosting method is shown as a function of boosting rounds for different values of k. It is not surprising that as the value of k increases (i.e., the constraint gets more restrictive) the number of training rounds for convergence decreases. However, if the value of $1/k$ is set to a rather large value the algorithm (i.e., weakening the weight constraint) may never converge. In this

example CDMABoost does not converge for $k < 5800$, which translates to the weight constraint: $w_i \leq 0.000172$.

4.6.4 Multiclass Classification Experiments

In this section we reports several experiments that we ran to evaluate the performance of the proposed multiclass boosting methods. Table 4.2 lists the multiclass datasets and their descriptions. The weak learners used in the first set of experiments were decision trees of size 5 and decision trees whose depth was restricted to be less than or equal to 8, that is, trees with maximum size of $2^9 - 1$. The performance of the algorithms Mu-MABoost, SAMME and Adaboost.M1 was compared and the obtained error rates are shown in Table 4.6. As can be seen, when the tree-size is small, Mu-MABoost achieves significantly better results (apart from dataset Car) than ADABoost.M1 and SAMME due to its optimal weak-learning condition. Compared with ADABoost.M1, the Mu-MABoost algorithm does not expect high accuracy from the weak classifiers and thus can further drive the test error down. SAMME,

Data set	Mu-MABoost	ADABoost-M1	SAMME
depth of decision trees$\leq$ 5			
Abalone	76.60	95.10	87.70
Connect-4	29.07	43.51	48.00
Car	6.20	2.00	3.00
Forest	29.07	38.26	40.18
Letter	31.62	61.72	64.52
Poker	40.40	43.47	51.93
depth of decision trees$\leq$ 8			
Abalone	73.90	75.10	78.70
Connect-4	27.81	30.56	39.54
Car	3.40	2.40	1.40
Forest	25.96	25.36	25.78
Letter	3.90	7.32	3.15
Poker	37.88	38.65	45.38

Table 4.6: The reported values are the test error rates of the multiclass classifiers in percentage after 500 rounds of boosting.

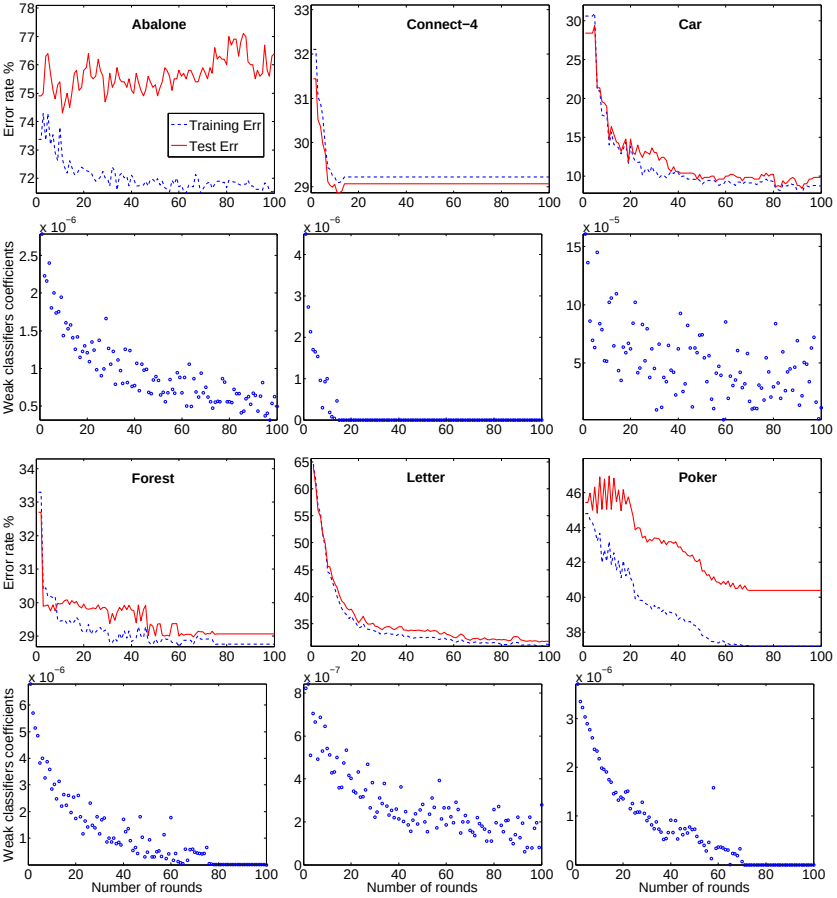

Figure 4.2: Plots of the rates at which Mu-MABoost drives down the training and test errors on various data sets over 100 rounds of boosting. Simple decision trees of size 5 are used as weak learners in Mu-MABoost. Since it expects an arguably low accuracy from the weak classifiers, the boosting can continue even with such a simple weak learner. The second and fourth rows illustrate the coefficients of the weak classifiers in the final ensembles.

on the other hand, uses even a weaker condition than Mu-MABoost. However, since this condition is not boostable, some of the weak classifiers in the SAMME ensemble in fact deteriorate the final classification accuracy and prevent the boosting method from driving the training error down. As the complexity of trees increases the gap between the performance of Mu-MABoost and ADABoost.M1 decreases since it is now more likely that a weak learner can satisfy the overly difficult demands of the Adaboost.M1 booster.

Figure 4.2 illustrates the training and test errors of the Mu-MABoost algorithm for different datasets. Moreover, the coefficients of the weak classifiers are also shown in the second and fourth column of figures in Figure 4.2. As boosting continues, the algorithm creates harder distributions and very soon the small-tree learning algorithms (CART in our case) may no longer be able to meet the excessive requirements of unnecessarily strong weak-learning conditions such as that of ADABoost.M1. However, the Mu-MABoost algorithm makes more reasonable demands that are easily met by CART. As can be observed, the coefficients of the weak classifiers are mostly greater than zero meaning that the weak-learning condition has been satisfied.

4.7 Conclusion and Discussion

In this chapter, a framework that can produce provable boosting algorithms was represented. Given a boostable classifier space $\mathcal{H}$, a provable boosting algorithm is a meta-algorithm that can provably drive the training error to zero by repeatedly calling a base learning algorithm, each time feeding it with a different distribution or weights over the training samples. The base learning algorithm returns a weak classifier selected from $\mathcal{H}$ and the boosting algorithm assigns a coefficient to it which is proportional to the weak classifier's performance. The final *strong* classifier is then the linear combination of these weak classifiers.

The proposed framework is suitable for designing boosting algorithms with distribution constraints, i.e., deriving boosting algorithms for applications where it is required to put more weight on some specific subset of examples than on others, either because these examples have more reliable labels or it is a problem requirement to have a more accurate hypothesis for that part of the sample space.

Several new boosting algorithms were derived from our proposed framework. Particularly, a sparse boosting algorithm (SparseBoost), which samples

only a fraction of examples at each boosting round and hence, efficiently reduces the required memory during the training process, was introduced. In fact this problem, i.e., applying batch boosting on successive subsets of data when there is not sufficient memory to store the entire dataset, was first discussed by Breiman in [Bre97], though no algorithm with theoretical guarantee was suggested. SparseBoost is the first provable batch booster that can (partially) address this problem. However, since our proposed algorithm cannot control the exact number of zeros in the weight vector, a natural extension to this algorithm is to develop a boosting algorithm that receives the sparsity level as an input. However, this immediately raises the question: what is the maximum number of examples that can be removed at each round from the dataset, while still achieving a $(1-\epsilon)$-accurate hypothesis?

We introduced the CD-MABoost algorithm as the first boosting methods that can take the quality of the datasets into account when several datasets with different quality are utilized to learn a particular task. The quality of a dataset depends on many factors including the level of feature noise and mislabeling noise. When the quality of the datasets are (qualitatively) given, this information can be incorporated in the learning process by restricting the importance or weights of the low quality samples. As shown in the experiments, while no other boosting methods (with convex loss function) can learn the binary classification task introduced in [LS10], CD-MABoost can easily learn the optimal hypothesis by restring the weights of the noisy samples.

The boosting framework derived in this work is essentially the dual of the online mirror descent algorithm. This framework can be generalized in different ways. Here, it was shown that replacing the Bregman projection step with the double-projection strategy, or as it was called approximate Bregman projection, still results in a boosting algorithm in the active version of MABoost, though this may not hold for the lazy version. In some special cases (MADABoost for instance), however, it can be shown that this double-projection strategy works for the lazy version as well. Our conjecture is that under some conditions on the first convex set (i.e., the convex set $\mathcal{K}$ in Lemma 4.8 which is assumed to be larger than the probability simplex $\mathcal{S}$), the lazy version can also be generalized to work with the approximate projection operator.

A new error bound is provided for the MADABoost algorithm that does not depend on any assumption. Unlike the common conjecture, the convergence rate of MADABoost (at least with our choice of η) is of $O(1/\epsilon^2)$. It is however, still an open question whether it is a tight bound or it can be improved.

Finally, the MABoost framework was generalized to the multiclass setting and was shown that this framework adopts the minimal weak-learning condition to boost the weak-learning space. That is, it is proven that it can employ very simple weak classifiers while still perfectly learning the underlying hypothesis or in other words, driving the training error down to zero. By using very weak classifiers in the ensemble, we restrict the complexity of the model and thus reduce the overfitting effect. This property, i.e., robustness against overfitting, is an essential requirement in our lipreading application where feature vectors are drawn from a high-dimensional space.

Chapter 5

Visual Voice Activity Detection

Voice activity detection (VAD) is a necessary stage in most speech-related applications such as speech recognizers (to identify which audio frames need to be processed) and human-machine interaction systems (to detect human activities). While using a reliable VAD may significantly improve their performance, it is not easy to guarantee the VAD accuracy especially in adverse conditions such as highly reverberant rooms, non-stationary speech-type background noise, etc.

As the technology advances, it is now becoming very cheap, both computationally and economically, to capture video streams. Thus, a natural solution to improving A-VAD systems is to utilize visual information as complementary information. However, most of the reported V-VAD systems in the literature suffer from speaker-dependency issues and inaccurate detection rates when used without audio features.

In this Chapter, by using SIFT features and the bag-of-words (BoW) model explained in Chapter 2 and the binary classifier developed in Chapter 4, we construct a V-VAD system that, first, is highly speaker independent and second, can achieve high accuracy (78% frame-based detection rate, on average). Moreover, we show that this system can be trained in a semi-supervised manner. As it is known that manually labeling speech boundaries in audio-visual data is not only a labor-intensive and time-consuming task,

but also subject to human errors and interpretations, having a system that can be trained in a semi-supervised manner is highly desirable. In order to obtain semi-supervised AV-VAD, we develop a learning algorithm that can detect noisy samples whose labels are randomly flipped. This algorithm obtains up to 95% detection rate on some datasets, which is very promising and perhaps can be used in a much wider range of applications.

5.1 Introduction to Utterance Detection

To efficiently use visual information, two challenging issues have to be addressed.

The first issue in V-VAD systems is the relatively high computational complexity of the feature extraction part. In general, all visual speech processing systems require a region-of-interest (ROI) from which visual features can be extracted. Visual feature extraction algorithms roughly fall into two categories: Those that use a crude estimate of ROI to extract visual features and those that utilize more advanced methods such as active appearance models [CET01] or active shape models [LTB96] to match an exact ROI location or an extract ROI contour [QWP11]. Clearly, the second category may yield more precise or informative visual features at the expense of higher computational complexity. In addition, both groups can benefit from an ROI tracking system to improve robustness [AMM+07], which again increases the computational complexity. Here we use the real-time algorithm presented in [DGFVG12] for mouth detection. Due to the computational efficiency of this algorithm, the amount of time spent for ROI extraction in our V-VAD algorithm is favorably small.

The second and even more complicated challenging issue of V-VADs is speaker variability. Speaker independence is a fundamental requirement in typical real-world V-VAD applications. In audio-based VADs, and more generally in audio-only speech recognition, speaker variability has been well studied. It has been shown that audio features (mainly MFCC for speech recognition and relative energy level and zero crossing rate for VADs) are highly speaker-independent. However, it is still a big challenge to develop a speaker-independent V-VAD and, more generaly, a lipreading algorithm that can cope with speaker variability (see Chapter 6 and [CHLN08]).

It is known that most commonly applied visual features such as DCT or PCA of the mouth region mainly represent speaker characteristics rather than

speech events. Thus, most of the currently available V-VADs cannot reliably work in the speaker-independent mode [MLS+13], [AM08] and [YNO09]. For instance, the performance of the V-VAD proposed in [AM08] drops from 97% to 72%, which is only 7% above randomness considering that 65% of the samples in their dataset are speech.

Here, we propose a V-VAD which is highly robust against speaker variability. Experiments show that the proposed V-VAD algorithm can obtain 78% frame-based detection rate on the GRID dataset, which is much higher than commonly used technology which employ Gaussian mixture models trained with DCT, PCA or LDA features extracted from video frames. Moreover, it is shown that this algorithm can be trained in a semi-supervised manner. This useful property can be exploited to automatically adapt this algorithm to new test conditions.

5.2 Supervised Learning: VAD by Using SparseBoost

Throughout this section, we develop a V-VAD by means of the SparseBoost algorithm listed in Algorithm 4.2. A SparseBoost classifier is trained in a supervised manner to classify speech and non-speech video frames.

From each ROI, a set of SIFT feature vectors were extracted. The SIFT feature vectors of all frames were then clustered into 300 groups. The set of cluster centers were considered to be a codebook with 300 codes. This codebook was used in a BoW model to construct 300-dimensional feature vectors. To reduce the computational complexity, the SIFT features were only computed for a subregion whose area was almost 1/5 of the detected mouth region and was a rectangle around the middle of the mouth. Since this limited region was sufficient to determine whether the mouth is open or closed it provided sufficient statistics to learn the desired hypothesis. This reduction in the ROI size improved the computational efficiency of the algorithm and helped the classifier to find a more accurate hypothesis by filtering out redundant information. Figure 5.1 depicts the ROI and the visual features used in our V-VAD algorithm.

To evaluate the proposed V-VAD algorithm, the GRID dataset explained in [CBCS06] was used in our experiments. Due to the large size of this dataset, we only employed the the audio-visual data of the first 16 speakers.

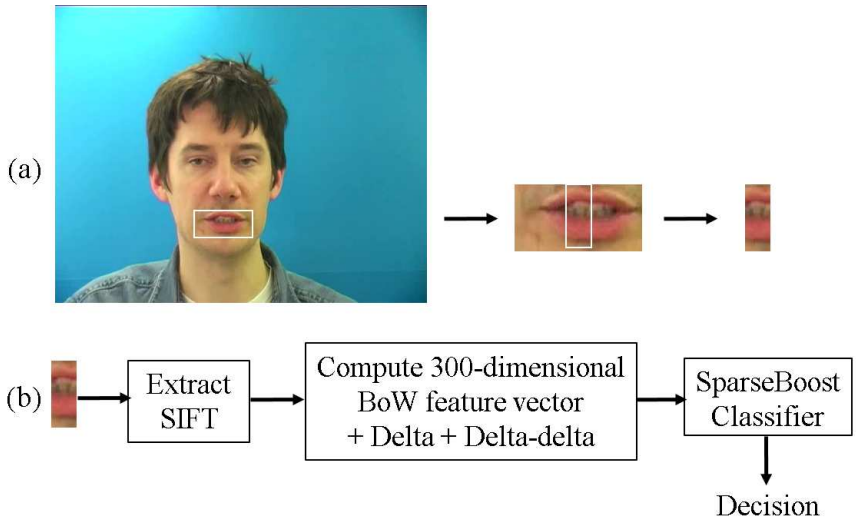

Figure 5.1: (a) The mouth region is automatically detected by means of the regression based random forest algorithm proposed in [DGFVG12]. However, only a small region from the middle of the mouth is effectively used in V-VAD. (b) SIFT features are extracted from the middle of the mouth. The set of extracted SIFT feature-vectors is mapped to a 300-dimensional BoW feature vector representing this frame. First and second-order derivatives of BoW features are also included in the feature set to construct the final visual feature vector. SparseBoost then takes this vector to make the decision on whether it is a speech or non-speech frame.

The recordings of the first 9 speakers, i.e. s1 to s9 are always included in the training data. We then applied the one-speaker-out cross-validation technique over the second set of speakers (s10-s16) to evaluate the proposed methods. At each round of cross-validation, the data from 6 of the second 7 speakers (s10 to s16) was added to the training data and one was left out for test. Each video sample of GRID data is 3 seconds long. To have an almost balanced dataset (i.e., almost the same number of speech and non-speech frames), only the last two seconds of each recording were used. This gave a dataset with 57% speech and 43% non-speech frames.

In the first experiment, the performance of GMM+DCT based V-VAD

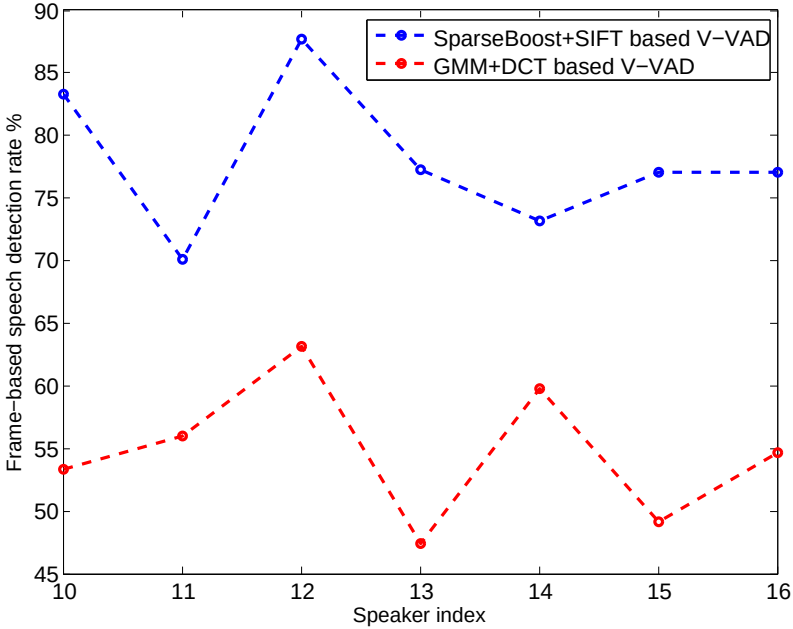

Figure 5.2: Comparison of the proposed approach (SparseBoost+SIFT) with GMM+DCT method.

and our proposed approach (called SparseBoost+SIFT) were compared. In the GMM+DCT based V-VAD approach, the forty highest energy DCT features and their first and second-order derivatives were extracted from ROIs. Two Gaussian mixture models, each with 4 Gaussian components, were then trained to model the speech and non-speech frames. This approach is commonly taken in the literature for both audio and visual based VADs [SKS99] [AM08]. We used simple decision trees depth limited to 5 as the weak learners in the SparseBoost algorithm. Figure 5.2 demonstrates the performance of these approaches for every speaker. As shown, the proposed method significantly outperforms the commonly used GMM+DCT based V-VAD approach. As seen, the GMM-based approach performs even worse than random guessing on speaker 12. This is due to the fact that DCT features are highly speaker-dependent and a generative method such as GMM cannot extract the relevant information from the dominant non-relevant information.

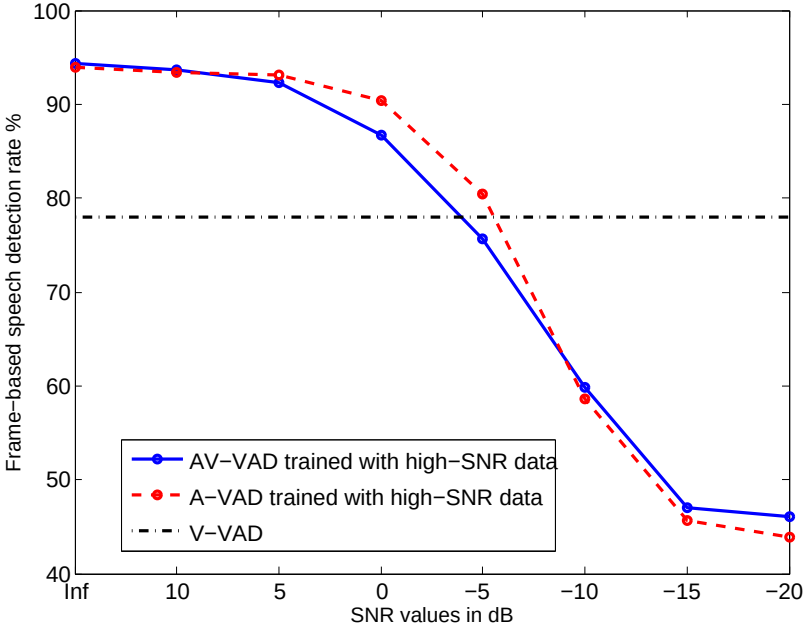

Figure 5.3: The accuracy of audio-visual VAD, A-VAD and V-VAD in percentage. A-VAD is trained with high-SNR audio data, containing only clean and 10 dB audio samples.

The same observation regarding the poor performance of DCT features in speaker-independent settings has also been reported in [AM08] and [Gur09]. Particularly in [Gur09] it was shown that the GMM+DCT method used to construct the silence model yielded very poor results and leaded to many deletions in continuous speech recognition. The mean accuracy of our method is $78\% \pm 2.24$, where 2.24 is the standard deviation calculated over the speakers. To the best of our knowledge, this method yields one of the best V-VAD systems reported in the literature in terms of both accuracy and the speaker-independence property. Given this robust V-VAD, the next natural step is to optimally combine it with audio-based VAD to obtain an AV-VAD that can accurately perform in adverse environments. As in V-VAD, we utilized the SparseBoost algorithm to train an A-VAD system. The 13-dimensional MFCC features and their first and second-order derivatives were extracted

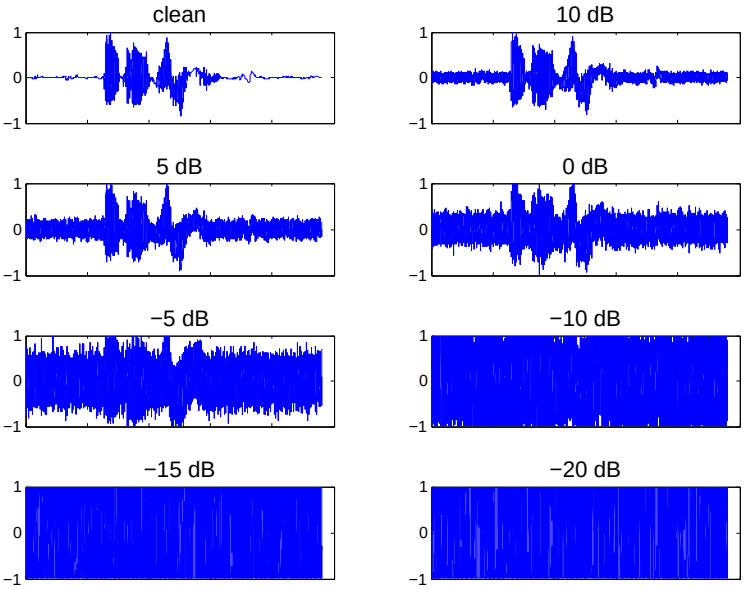

Figure 5.4: Audio signals with various signal-to-noise ratios. In -20 dB, signal is completely dominated by noise.

from frames with 65 ms duration and 25 ms overlap. That is, similar to video data, the audio frame-rate is 25 frames per second.

In the first set of experiments, we trained the A-VAD with a training set where half of the audio signals were clean and the other half were 10 dB audio signals (constructed by adding additive white noise to them). The audio and visual VADs were independently trained and their decisions were then fed to an ensemble of decision trees, with 100 trees, to make the final decision. This fusing strategy is known as the late decision fusion method. As seen in Figure 5.3, while AV-VAD slightly improves the A-VAD system in low-SNR regions, in most SNR values it performs worse than or on a par with the A-VAD, which considering its higher memory and computational cost, is not acceptable. More importantly, even in the low-SNRs where the AV-VAD works slightly better than A-VAD, its performance is still far worse than the simple V-VAD system. This is an indication that the visual information is

largely disregarded in the AV-VAD. This happens due to the fact that in our training data where audio signals are either clean or have 10 dB SNRs, A-VAD system is much more accurate than V-VAD. Thus, the final classifier which takes the decisions of audio and visual VADs as input, learns to ignore the V-VAD decision.

To overcome this problem, in the next experiment we trained the A-VAD with a training set containing audio signals at various SNRs (by adding a random amount of white noise to each utterance). Figure 5.4 shows audio signals at various SNRs. It is clear that including -20 dB audio samples in the training set will not result in a A-VAD that can accurately work in -20 dB, simply because in -20 dB it is even almost impossible for a human to detect speech frames. By using this mixed training data, however, the A-VAD system learned to assign very low scores[1] to the samples with low-SNRs, or generally to the samples that it cannot classify. In other words, it learned to detect when it fails. The final classifier then learned to use the V-VAD decision for samples whose A-VAD scores were too small.

Figure 5.5 depicts the accuracy of the AV-VAD when training data contains audio samples at various SNRs. As seen, this AV-VAD shows significant improvement over the AV-VAD trained with high-SNR audio samples. This amount of improvement, however, may still not justify its higher computational complexity compared with a simple A-VAD system. To further improve the proposed AV-VAD, one may note that in the late fusion strategy, the whole information regarding the audio (video) modality is compressed into one single value, i.e., the output of the A-VAD (V-VAD) and is passed to the final classifier. This gives very limited degrees of freedom to the final classifier to fuse audio and visual information.

To circumvent this problem, instead of training individual audio and visual VADs, the SparseBoost classifier was trained with super-vectors constructed by concatenating audio and visual features. By observing both audio and visual features simultaneously, the classifier could explore their complementary information and find a hypothesis that could fully take advantage of the both modalities.

The blue curve in Figure 5.5 demonstrates the performance of the AV-VAD with the early fusion strategy and trained with mixed audio data. As shown, it outperforms the other methods almost at all SNRs and its accuracy

[1] The output of the A-VAD is assumed to be a continuous value in $[-1, 1]$. A value close to zero indicates low confidence in the decision.

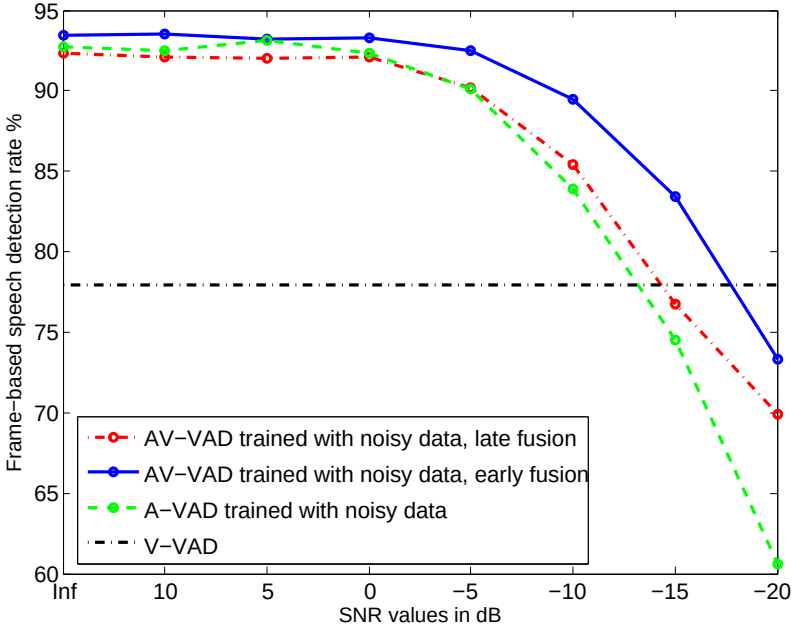

Figure 5.5: The accuracy of audio-visual VAD for both late and early fusion, A-VAD and V-VAD in percentage. A-VAD is trained with mixed audio data, containing audio samples at various SNR values. AV-VAD with early fusion outperforms other methods.

is higher than that of the visual-only or audio-only VAD algorithms.

5.3 Semi-supervised Learning: Audio-visual VAD

Labeling data is an expensive, time-consuming task while in many applications, unlabeled data is cheap and abundant due to the internet. Moreover, in some applications such as applying VAD to youtube clips, many environmental variables such as, microphone type, camera resolution and background noise are changing from one video to another. Usually in these situations only unlabeled data are available to adapt a VAD to this varying environment.

Recently, there has been a growing interest in developing unsupervised VAD systems [GSM13, KR13, YYDS11]. In this work, we further explore this line of research by taking advantage of the fact that audio and visual streams give two very different (and conditionally independent) representations of the same underlying phenomenon ,i.e. speech. With this observation, our problem can be cast into now the classical two-view unsupervised training framework presented in [BM98]. Due to this framework, two learning algorithms are applied on each view of the data (audio and visual) separately and then each algorithm's predictions over unlabeled data are used to enlarge the training set of the other algorithm.

It was shown in [BM98] that when a data description can be partitioned into two distinct views, any initial weak predictor (trained with a small set of labeled data) can be boosted to arbitrary high accuracy using unlabeled examples only by co-training. The two important assumptions based on which this celebrated result holds are: First, the underlying hypothesis is learnable in the presence of labeling noise and second, the two views of data are conditionally independent, given the label. While the latter assumption is satisfied in our audio-visual application, the first assumption largely depends on the employed learning algorithm. Some learning algorithms are more sensitive to labeling errors than others. For instance, it is known that for each learning algorithm with a convex loss function, there is a labeling noise with a particular distribution so that in the presence of this noise the algorithm fails to converge to a hypothesis better random guessing [LS10].

In this work, we develop a learning algorithm, called Ro-MABoost, which can detect and remove mislabeled samples from the training data with high accuracy. We show that this algorithm can detect up to 95% of the labeling errors on some datasets, as long as the errors are randomly induced or encountered from the learning algorithm point of view. That is, when there is no systematic labeling error or in other words, when the labeling error does not follow any pattern.

Since Ro-MABoost can be used to learn the underlying hypothesis from a noisy dataset, it is a suitable choice for our semi-supervised application in which the audio and visual training sets are recursively labeled by non-perfect learning algorithms. The proposed semi-supervised AV-VAD training algorithm is outlined in Algorithm 5.1.

In the following, we explain in detail the Ro-MABoost learning algorithm

used in Algorithm 5.1.

Algorithm 5.1: Semi-supervised audio-visual VAD training

Input: Labeled training data $\mathcal{D}_{\text{l}} = (\mathcal{D}_{\text{l}}^{A}, \mathcal{D}_{\text{l}}^{V})$,
unlabeled training data $\mathcal{D}_{\text{u}} = (\mathcal{D}_{\text{u}}^{A}, \mathcal{D}_{\text{u}}^{V})$,
convergence threshold ϵ and $\triangle E = \infty$.

Supervised stage: Train A-VAD with $\mathcal{D}_{\text{l}}^{A}$ and label $\mathcal{D}_{\text{u}}$ with it,
set $E_1 =$ error rate of A-VAD.

While $\triangle E > \epsilon$ **do**

(a) **V-VAD**: Train Ro-MABoost with $\mathcal{D}_{\text{l}}^{V} \cup \mathcal{D}_{\text{u}}^{V}$ and
output indices of mislabeled samples $\mathcal{I}$.

(b) **Omitting mislabeled data**: Remove $\{\mathbf{x}_i \mid i \in \mathcal{I}\}$ from training data,
$\mathcal{D} = \mathcal{D}_{\text{u}} \setminus \{\mathbf{x}_i \mid i \in \mathcal{I}\}$

(c) **A-VAD**: Train Ro-MABoost with $\mathcal{D}_{\text{l}}^{A} \cup \mathcal{D}^{A}$ and output h_A.

(d) **Re-labeling**: Re-label $\mathcal{D}_{\text{u}}$ with h_A.

(e) **Stopping Criterion**: $E_2 =$ error rate of h_A,
set $\triangle E = E_1 - E_2$ and $E_1 = E_2$.

End

Output: Ouput h_V and h_A.

5.3.1 Ro-MABoost

As mentioned in Chapter 4, the concept of boosting emerged as an answer to the question whether a weak learner can be converted to a strong learner with arbitrary accuracy. A boosting algorithm is a meta-algorithm that generates many weak learners with slightly better-than-random performance. The final strong classifier is the ensemble of these weak classifiers. At each round of training, the algorithm concentrates on the samples that are wrongly classified in the previous steps and aims to find a hypothesis that accurately describes them. However, when some training samples have wrong labels, this learning scheme may badly fail.

Algorithm 5.2: Robust Mirror Ascent Boosting (Ro-MABoost)

Input: $\mathbf{w}_1 = [\frac{1}{N}, \ldots, \frac{1}{N}]^\top$, $\mathbf{z}_1 = [\frac{1}{N}, \ldots, \frac{1}{N}]^\top$ and $\mathcal{I} = \{\}$

For $t = 1, \ldots, T$ **do**

(a) Train classifier with $\mathbf{w}_t$ and get h_t,
let $\mathbf{d}_t = [-a_1 h_t(\mathbf{x}_1), \ldots, -a_N h_t(\mathbf{x}_N)]$ and $\gamma_t = -\mathbf{w}_t^\top \mathbf{d}_t$.

(b) Set $\eta_t = \frac{\gamma_t}{N}$ and $\mathcal{I}^c = \{1, \ldots, N\} \setminus \mathcal{I}$

(c) **If** $|\mathcal{I}| < N\varepsilon$:
For $i \in \mathcal{I}^c$ **do**:
$\theta_i = \frac{a_i \sum_{t'=1}^{t} \eta_{t'} h_{t'}(\mathbf{x}_i)}{\sum_{t'=1}^{t} \eta_{t'}}$ # The margin of the i-th example
If $\theta_i < \Theta$
$\mathcal{I} = \mathcal{I} \cup i$
End
End
End

(d) Set $\mathcal{S} = \{\mathbf{w} \mid w_i \geq 0 \ \ \forall i \in \mathcal{I}^c, w_i = 0 \ \forall i \in \mathcal{I}, \ \sum_{i=1}^{N} w_i = 1\}$

(e) Project onto $\mathcal{S}$: $\mathbf{w}_{t+1} = \underset{\mathbf{w} \in \mathcal{S}}{\operatorname{argmin}} ||\mathbf{w} - \mathbf{w}_t - \eta_t \mathbf{d}_t||_2^2$

End

Output: The final hypothesis $f(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=1}^{T} \eta_t h_t(\mathbf{x}) \right)$.

The effect of random label noise on boosting can be intuitively explained as follows: Assume the data is separable. As the number of training rounds increases, the algorithm assigns higher and higher weights to *hard-to-classify* examples, which in this case are wrongly labeled samples. In ADABoost for instance, the weights of noisy samples increase exponentially fast with respect to the number of training rounds. That is, after a few rounds, the weights of correctly-labeled samples are very small compared with those of mislabeled ones, resulting in weak learners that fit the noisy samples.

To overcome this deficiency, we propose a give-up strategy to omit noisy samples during the training process. The algorithm takes two hyperparameters as inputs: First, the noise rate, which is the amount of noise in data and second, the margin threshold which is the minimum margin of the samples that are still considered to be correct. Given these two parameters,

at each round of training, the algorithm generates a weak classifier and removes the samples whose margins are smaller than the margin threshold. The removal process continues until the total number of removed samples reaches a threshold (computed from the noise rate). From this stage on, the algorithm continues to generate a weak classifier at each round, however, without removing any more samples from the training data. It finally returns the ensemble of the weak classifiers as the final classifier, when the predefined number of classifiers T are reached. This gives a robust version of the MABoost algorithm, called Ro-MABoost, which can handle noisy datasets.

Let $\{(\mathbf{x}_i, a_i)\}, 1 \leq i \leq N$, be N training samples, where $\mathbf{x}_i \in \mathcal{X}$ are feature vectors and $a_i \in \{-1, +1\}$ are labels. Assume $h \in \mathcal{H}$ is a real-valued function mapping $\mathcal{X}$ into $[-1, 1]$. Denote a distribution over the training data by $\mathbf{w} = [w_1, \ldots, w_N]^\top$ and define a loss vector $\mathbf{d} = [-a_1 h(\mathbf{x}_1), \ldots, -a_N h(\mathbf{x}_N)]^\top$. Assume the rate of labeling noise is ε and the margin threshold below which a sample is considered to be noisy is Θ. Using this notation, the Ro-MABoost is outlined in Algorithm 5.2. Set $\mathcal{I}$ in that algorithm keeps the indices of noisy samples and $\mathcal{S}$ is a subspace of the probability simplex in which only some of the dimensions can be nonzero (those that correspond to correct samples). Choosing $\mathcal{S}$ as defined in Algorithm 5.2 guarantees that the weights assigned to detected noisy samples are zero, resulting in excluding them of the training process.

Ro-MABoost can be described as follows: Assume there are 200 weak classifiers in the final ensemble, i.e. $T = 200$. These weak classifiers can be seen as very simple rules of thumb that roughly classify samples with accuracy just slightly better than random. At each round, Ro-MABoost removes the samples that are inconsistent with most of the already generated rules of thumb. For instance, by properly setting Θ, after generating 100 rules of thumb, Ro-MABoost starts to remove the samples that are wrongly classified by 90% of these classifiers. However, an important requirement in order to obtain reasonable results is to ensure that the weak classifiers used in Ro-MABoost are in fact very weak, otherwise, they even may fit the noisy samples. In our implementation, we used decision stumps (decision tree with only one node) as weak classifiers.

5.3.2 Experiments on Supervised & Semi-supervised VAD

In the first set of experiments, the accuracy of Ro-MABoost in detecting noisy samples was evaluated. Six of the binary datasets described in Table 4.1 were

employed in these experiments. The algorithm was tested in the presence of 3 different noise rates (ε): 10%, 20% and 30%. At each experiment, we randomly flipped a fraction of labels according to the given noise rate ε and used this noisy dataset to train a classifier by using Ro-MABoost. Throughout the training process, Ro-MABoost detects ε percentage of the samples as noisy samples. Table 5.1 reports the percentage of the identified noisy samples by Ro-MABoost which in fact were mislabeled samples, i.e. the accuracy of Ro-MABoost in detecting mislabeled samples. The margin threshold

Data set	**10% noise**	**20% noise**	**30% noise**
Breast cancer	95.71	93.57	84.76
German-credit	79.00	69.50	57.00
Votes-84	95.45	94.25	86.92
Pima-diabetes	57.14	58.44	46.08
Thyroid-disease	96.50	94.25	87.50
Sonar	52.38	66.66	40.32

Table 5.1: The accuracy of Ro-MABoost in detecting the mislabeled samples in the presence of various percentage of labeling noise reported in percentage. For instance, in the breast cancer dataset, Ro-MABoost can detect 95.71% of the mislabeled samples correctly.

used in these experiments was adaptive and was set to 0.2 times the mean margin (average of the margins of all samples) at each round. As seen in Table 5.1, Ro-MABoost achieves impressively high detection accuracy on most datasets. On 3 out of the 6 datasets, Ro-MABoost can detect more than 95% of mislabeled samples in the presence of 10% labeling noise and obtain more than 84% detection accuracy when the noise rate is as large as 30%. The success of Ro-MABoost in detecting mislabeled samples suggests further applications of this algorithm including search result improvement in search engines by detecting irrelevant results and improving the quality of training data.

In the next set of experiments we evaluated the performance of the semi-supervised AV-VAD algorithm listed in Algorithm 5.1. In this test, utterances of the speaker 12 were considered test data and the rest of the first 16 speakers to be training data. 1.25% of samples were randomly selected to construct the labeled training data and the remaining samples (i.e., 98.75%) were taken as unlabeled training data $\mathcal{D}_u$. The audio training set was a mixed of audio

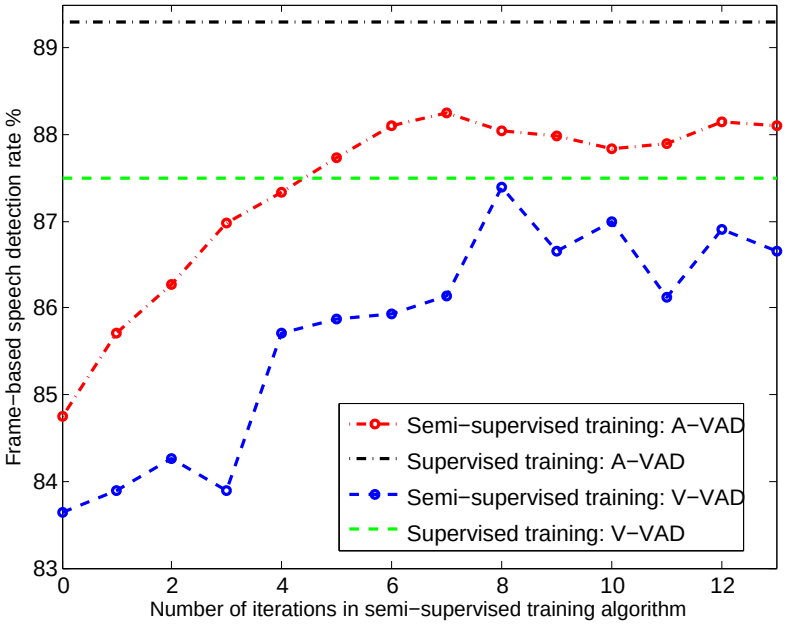

Figure 5.6: The accuracy of semi-supervised A-VAD and V-VAD in percentage. After 7 rounds of iterations, A-VAD reaches 88.25% detection rate which is only 1% lower than A-VAD accuracy obtained by supervised training.

signals at various SNRs (random amount of white noise was added to each utterance) and the audio test set was the audio recordings from speaker 12 with additive noise so that their SNR values were around 0 dB. Figure 5.6 demonstrates the test error of A-VAD and V-VAD over 13 iterations of training. The test error of round 0 corresponds to the accuracy of the initial A-VAD (which is trained on a given small set of labeled data). The Horizontal lines in Figure 5.6 show the accuracy of A-VAD and V-VAD over the test set when 100% of training samples were labeled and used in training.They work as reference points for our unsupervised training. The aim is to get as close as possible to these optimal upper bounds. As seen, the optimal A-VAD accuracy is 89.30%. By using unsupervised training we obtained 88.25% detection rate, which is only slightly lower than that of the fully labeled case. The V-VAD classifier also obtains up to 87.4%, which is only 0.1% lower than 87.5% accuracy of

the fully labeled case. Moreover, Algorithm 5.1 seems to converge after a relatively small number of iterations since the performance did not demonstrate significant improvement after the 8th iteration.

The only requirement of our semi-supervised AV-VAD training is to have an initial A-VAD. This A-VAD can either be obtained by training a classifier with a small set of labeled data (which in this case is better categorized as an example of semi-supervised training) or by simply using a threshold-based A-VAD (using energy as the main feature) as an initial A-VAD. Both interpretations can be of interest to practitioners. Particularly, in applications where adapting the AV-VAD to time-varying environments is a requirement, the already available A-VAD can be considered an initial A-VAD and Algorithm 5.1 can be used for adaptation.

5.4 Conclusion

Throughout this Chapter, we developed a V-VAD system that is highly speaker-independent and can achieve high accuracy. The proposed V-VAD method is obtained by training a MABoost classifier (described in Chapter 4). The final V-VAD is an ensemble of 4000 very simple decision trees (with depth less than 5). Even though the dimensionality of the feature vector space is fairly large (900), the proposed V-VAD achieves low generalization error due to the robustness of MABoost to overfitting.

We developed a robust version of the MABoost algorithm which can detect mislabeled samples in a training set. We showed that in some datasets, it can detect up to 95% of mislabeled samples when the labels are independently flipped with some small probability, i.e., the random label noise is not systematic (and thus not learnable).

Furthermore, we derived a semi-supervised AV-VAD framework that can be used to either adapt a pretrained AV-VAD to a new environment or to train an AV-VAD system over an unlabeled dataset by using an initial threshold-based A-VAD. This framework is based on the co-training algorithm proposed in [BM98] and our mislabeled sample detection algorithm, Ro-MABoost. In the experiments we showed that by using this framework, both A-VAD and V-VAD obtain detection accuracies close to that of the fully labeled case where all training samples are labeled.

Chapter 6

Lipreading with High-Dimensional Feature Vectors

This Chapter is devoted to developing a lipreading system based on the feature sets and the learning methods introduced in previous Chapters. In this system, the 3D-SIFT features are extracted from each video. These features are then used in bag-of-words (BoW) models to generate a 4000-dimensional BoW descriptor per frame. This high dimensional data is directly utilized to train a multiclass classifier by Mu-MABoost. This system is shown to be reliable in the sense that it is highly speaker-independent and outperforms conventional algorithms.

Similar to [HES00], where the probability posteriors generated by an ANN were used as audio features (known as Tandem features) in an GMM-HMM based speech recognizer, a new set of visual features are proposed wherein output posterior probabilities of the Mu-MABoost classifier are added to COBRA-selected ISCN features, introduced in Chapter 3. The new feature set is then used to train GMM-HMMs. Using this new feature set further reduces the variance of the V-ASR over different speakers and obtains 70.5% and 63.5% accuracy on Oulu and GRID datasets, respectively. To the best of our knowledge, These recognition rates are the highest rates reported

in the literature for these datasets when V-ASR is evaluated in the speaker-independent setting.

6.1 Introduction to Visual Speech Recognition

It is a known fact that speech perception is a multi-modal phenomenon. Audio speech, visual speech (lipreading), facial gesture, body pose, etc., all convey relevant speech information. In most scenarios, however, audio and visual signals are sufficient input for a human listener to perceive and understand speech.

Following this observation many researchers aimed to fill the gap between human level performance and the performance of A-ASR in real world applications by integrating additional visual speech information with audio speech. This, however, turned out to be too optimistic. The first attempts in this direction [PNLM04, and references there in] clearly showed that visual cues are highly speaker dependent and show large variations in different illumination conditions. That is, commonly used visual features (appearance features such as DCT, PCA and LDA and model based such as active shape model (ASM) [DL00] and active appearance model (AAM) [MCB+02, CET01]) are not reliable to work in a speaker-independent mode. This poses two questions: (I) A more theoretical question regarding the formal definition of reliability[1] which has been investigated in [PSSD06, RCB97] and (II) a practical question on how an unreliable recognizer (V-ASR) can be optimally combined with a reliable recognizer (A-ASR) in order to obtain an overall performance improvement.

A commonly used approach to tackle the second problem is through assigning a reliability weight (an exponent weight) to the probability distribution of each information stream. This approach has been empirically shown to be effective [WSC99], [CR98], [CDM02], [DL00] in audio-visual speech and speaker recognition, especially in the presence of varying additive acoustic noise. From the theoretical standpoint, it was shown in [PSSD06] that, under certain simplifying assumptions, proper choice of exponent weights can minimize the variance of mismatch error between true and estimated distributions,

[1] Note that reliability is a property corresponding to the features and the amount of mismatch error between the distributions of the training and test data. For instance, an A-ASR system trained over a noisy dataset is a weak recognizer while it is still considered to be reliable due to its consistent performance over different speakers.

though as we discussed in Chapter 7, this may not necessarily result in lower error rate. However, due to the strong speaker- and illumination-dependence of visual features, these weighting schemes should be able to automatically adapt themselves in order to cope with the quality variations of visual modality. This, however, is an inherently challenging problem due to the fact that the quality of visual features should be assessed in an unsupervised manner.

To sidestep this problem, in Chapter 7 we derived a weight estimation algorithm by maximizing the AUC criterion. Instead of optimizing the performance of the recognizer for a particular test condition, the proposed approach yields a set of weights minimizing the expectation of the error over different test conditions. The proposed fusion algorithm is a general approach that can be applied to the model presented in this chapter, as well. However, due its pessimistic nature, this robustness comes at the expense of some performance degradation of the AV-ASR at the high SNR regimes.

Another approach to improving the reliability of the V-ASR systems is by using speaker- and illumination-invariant features. A reliable V-ASR is a classifier whose performance shows little variation over different speakers. A big step towards this direction was taken by Zhao et.al. in [ZBP09], where they considered the whole video sequence of an utterance a volume and calculated spatio-temporal local binary pattern (LBP-TOP) features in order to classify utterances. LBP features are binary features known to be gray-scale and rotation invariant [OPM02]. When calculated across XY,YT and XT plans, they result in LBP-TOP features which, in addition to invariance properties inherited from LBP features, encode the time-dynamics of visual speech. As they demonstrated, the temporal patterns have a highly discriminative power since they are less sensitive to speakers and more related to underlying speech. However, they applied two further steps to extract low-dimensional equal-size feature vectors from videos. First, even though different utterances have different length (i.e., various number of frames), they are divided into the same number of block volumes to get the same number of features. Second, ADABoost is employed to select a small subset of informative features. From our point of view, these two steps are unnecessary and perhaps reduce the invariance properties of LBP-TOP features.

Conventional visual features are highly speaker-dependent and more suitable to encode the identity of a speaker rather than visual speech. Normalizing visual features, thus, is a reasonable idea in V-ASR systems in order to improve the speaker-invariance property of the recognizer. For instance, in [ZZP11] it is proposed to first learn a deterministic function that maps a

curve (which in their work is a low-dimensional manifold) into the image space and then use this function to normalize raw visual data. They showed using this method in conjunction with LBP-TOP features resulted in overall 10% performance improvement in a 10-phrase recognition task. Other normalization methods such as Hi-LDA [PNLM04] and z-score normalization [NC09] are also commonly used in the literature. An extensive comparison between different normalization methods for AAM and DCT feature sets has been reported in [LTH+10]. However, these methods do not completely solve the inter-speaker variation problem and the variation of the V-ASR performance over different speakers is still substantially large.

Perhaps, the main reason of the sensitivity of conventional visual features to speaker's identity is because they are low-level global features. According to the definition, global features are features that are dependent on the values of all pixels. For instance, DCT, LDA or PCA are global features since they are directly extracted from pixel values by applying a linear transformation to an ROI. The main drawback of global features such as LDA is that since ROIs usually convey a significant amount of irrelevant information (i.e., facial characteristics of the speaker), these features are highly noisy and thus difficult to learn from. Similar to LDA, AAM and ASM methods are also global features because even though they further process ROI to extract lip active appearances (active shapes in case of ASM), a new feature vector is still extracted from an ROI by applying a linear transformation to an ROI.

In Chapter 3, we showed that using ISCN features improves the V-ASR performance. However, even though ISCN features are the results of applying several non-linear local wavelet transforms to an ROI, they still cannot obtain a sufficient level of robustness, necessary for V-ASR applications. To address this problem, we use a discriminative approach and train a MABoost classifier with visual features. Posterior probabilities of this classifier are then added to ISCN features to construct the final set of visual features which are used to train GMM-HMMs.

The key contributions of this work are threefold:

1. **Multiple color spaces**: In order to achieve illumination-invariance and increase discriminative power, video frames are first transformed to 5 different color spaces: opponent color space which is invariant to changes in light intensity, color-invariant color space which is scale-invariant with respect to light intensity, *r* and *g* chromatic components of the normalized RGB color which are known to be scale-invariant and

finally, the normal RGB color space and the gray color space. The invariance properties of each of these color spaces and the discriminative power of their corresponding scale-invariant feature transform (SIFT) features [Low04] have been explored in [vdSGS10]. It was shown there that using a combined set of features extracted from these color spaces results in a significant performance improvement and enhances the robustness of object classification against illumination variations.

2. **3D-SIFT + Bag-of-Words**: Following the idea of Zhao et. al. [ZBP09] to use spatio-temporal features, we also employ 3-dimensional (X,Y and time, T) features to represent visual speech. However, instead of LBP-TOP, we employ 3D-SIFT introduced in [SAS07] in order to enjoy the invariance properties of the popular SIFT features. From a problem-oriented standpoint, the most important invariance property of SIFT in a lipreading task, where speakers have different lip and mouth characteristics and sizes, is its scale-invariance property. 3D-SIFT features are extracted from each frame and for each of the 5 color spaces mentioned above, i.e., 5 sets of feature vectors per frame. Since each frame may have various numbers of key points, the number of 3D-SIFT feature vectors representing each frame is different from the others. To obtain fixed-length feature vectors per frame, the bag-of-words model introduced in [SZ03] is employed. Using BoW is perhaps the most important step in this work in order to achieve speaker-invariant features.

3. **Multiclass Boosting**: Finally, for classification, a learning algorithm needs to be used that has the following properties: (I) Due to the high-dimensionality of visual feature vectors, the learning algorithm should neither have many free parameters nor be too complex in order to be resistant to over-fitting. (II) It should be powerful enough to learn the underlying complex hypothesis. (III) It should be able to directly address the multiclass classification problem. The common approach to dealing with multiclass settings is to reduce them to multiple binary classification problems. This, however, significantly increases the computational complexity in our problem where five high-dimensional feature sets are used to train five classifiers for digit classification and five 26-class classifiers for letter recognition.

   We use the multiclass setting of MABoost algorithm (called Mu-MABoost) presented in Chapter 4 and demonstrate that it satisfies the above requirements. Mu-MABoost has several appealing properties.

First, it was shown in Chapter 4 that Mu-MABoost requires minimal conditions on weak-classifiers to achieve high accuracy on training data. That is, weak classifiers constituting the final strong classifier are chosen from a very simple hypothesis space, in our case, from the class of decision trees with a depth less than 5. Using such simple classifiers will minimize the risk of over-fitting. Second, by directly formulating the multiclass classification problem, it largely reduces the computational complexity, making it a method of choice for high-dimensional data.

4. **ISCN + Posterior Probabilities:** Due to the success of HMMs in A-ASR systems [RJ93], HMMs and more generally deep Bayesian networks [CH97, LT97, SLS+05] were largely applied in V-ASR to capture the time-dynamics (temporal information) of visual speech. We also adapt HMMs in our final recognizer framework in order to better take the time-series aspect of speech into account. The visual features used to train the final GMM-HMM based lipreading system are posterior probabilities which are the output of Mu-MABoost and COBRA-selected ISCN features introduced in Chapter 3. This hybrid feature set obtains very promising performance and outperforms other feature extraction methods on the GRID and Oulu datasets.

6.2 Color Spaces

It is shown in [vK70] that changes in illumination can be modeled by a linear transformation (diagonal mapping or von Kries model). Thus, different color spaces for representing an image may present different invariance. As it is extensively explained in [vdSGS10], based on von Kries model and the von Kries offset model, four types of changes in an image can be described. These four image transformations due to the light change are light intensity change, light color change, light shift change and finally, light color and shift change.

Proper color spaces, however, show invariance properties against some of these changes. The color spaces used in this work are as follows:

1. **Opponent color space**: It is defined as

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix} \tag{6.1}$$

   where R, G and B are the three channels of the RGB color space. In this space, O_1 and O_2 are shift-invariant with respect to light intensity and O_3 does not have any invariance property.

2. **Color-invariant color space**: It is defined as

$$\begin{pmatrix} H \\ \\ C \end{pmatrix} = \begin{pmatrix} \frac{0.3R+0.04G-0.35B}{0.34R-0.6G+0.17B} \\ \\ \frac{0.3R+0.04G-0.35B}{0.06R+0.63G+0.27B} \end{pmatrix} \tag{6.2}$$

   H and C are the two color-invariant channels used to extract C-SIFT [AF06] (to be precise, in [AF06] only H channel was eventually used to obtain C-SIFT features). As explained in [vdSGS10], H and C are scale-invariant with respect to light intensity but not shift-invariant.

3. **rg color space**: It is the normalized RGB color model

$$\begin{pmatrix} r \\ g \end{pmatrix} = \begin{pmatrix} \frac{R}{R+G+B} \\ \frac{G}{R+G+B} \end{pmatrix} \tag{6.3}$$

   Due to the intensity normalization, r and g are scale-invariant with respect to light intensity changes, shadows and shading.

4. **Gray scale**: It is the most commonly used color space in V-ASR systems due to its simplicity (mono-channel) and can be written as

$$g = 0.2989R + 0.5870B + 0.1140G \tag{6.4}$$

   Gray scale has no invariance properties.

5. **RGB color space**: The last color space used in this work is the RGB color space which obviously has no invariance properties.

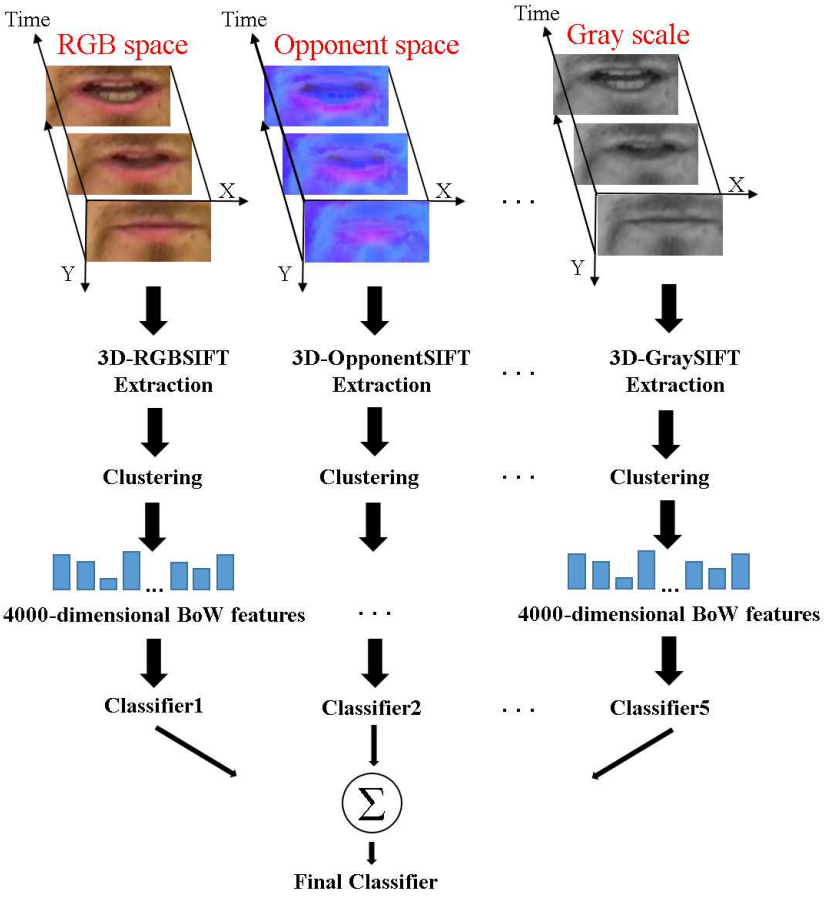

Figure 6.1: Visual speech-unit classifier. In the first step, video data are transformed to 5 color spaces: RGB, opponent, rg, gray and color-invariant. 3D-SIFT features are extracted from each representation and clustered into 4000 classes. BoW features, which are the frequency of occurrence of each of these classes, are then computed for each video sample and employed to train 5 classifiers, one for each color space. The final classifier is the ensemble of these 5 classifiers.

6.3 Visual Features

Despite the fact that many visual feature sets have been proposed for V-ASR systems (see [ZZHP14] and [PNLM04]), none of them have gained wide acceptance (as MFCCs in A-ASR systems) in real-world applications. Recently, however, several spatio-temporal features have appeared which are shown to be promising in both video classification [SAS07] and lipreading [PGC08] [ZBP09]. Our proposed method may also be considered to go along this line with an additional bags-of-words step which results in a sparse feature vector (suitable for decision tree classifiers used as weak classifiers in Mu-MABoost).

In this work, video samples are first represented in 5 different color spaces introduced in the previous section. For each representation, a set of 3D-SIFT features[2] are extracted from the video data and be used in a BoW model with 4000 words to construct the final 4000-dimensional feature vectors. This approach yields 5 feature sets, each containing 4000-dimensional feature vectors. Each of these feature sets are then used to train a multiclass classifier for speech recognition. This procedure is shown in Figure 6.1.

6.4 Multiclass Classifier

In order to fully benefit from the invariance properties and sparsity of the extracted features, we need a learning algorithm that (I) can exploit the sparsity of the feature vectors to reduce the computational complexity of the learning phase and (II) aggressively learn the relevant information in the training data with minimum overfitting. The Mu-MABoost algorithm presented in Chapter 4 is in fact satisfies both of these requirements. First, by using shallow decision trees as weak learners, it can benefit from the sparse structure presented in the feature vectors and second, since it is a provable boosting algorithm, it guarantees to drive training error down to zero in a finite number of iterations.

In this work, the Mu-MABoost algorithm with the Euclidean distance as the Bregman divergence is used to train visual digit and letter classifiers. This algorithm is listed in Algorithm 6.1. In Algorithm 6.1, $\text{vec}(\mathbf{W})$ stands for the vectorized version of matrix $\mathbf{W}$ (by column-wise concatenation), $\text{Tr}(.)$

[2] 3D-SIFT descriptors are the direct generalization of the popular SIFT features. A brief explanation of 3D-SIFT is given in Chapter 2.

denotes the trace operator, N is the number of training examples and K is the number of classes.

Algorithm 6.1: K-class MABoost (Mu-MABoost)

Input: $\mathcal{R}(\mathbf{X}) = \mathrm{Tr}(\mathbf{X}^\top \mathbf{X})$,
$\mathbf{W}_1$ and $\mathbf{Z}_1$ with elements $W^1_{i,j} = Z^1_{i,j} = \frac{1}{N(K-1)}$

For $t = 1, \ldots, T$ **do**

(a) Train a classifier with $\mathbf{W}_t$ and get h_t, set $\mathbf{D}_t$ elements as in (4.18) and $\gamma_t = -\mathrm{Tr}(\mathbf{W}_t^\top \mathbf{D}_t)$.

(b) Set $\eta_t = \frac{\gamma_t}{(K-1)N}$

(c) Update: $\mathbf{Z}_{t+1} = \mathbf{W}_t + \eta_t \mathbf{D}_t$

(d) Project onto $\mathcal{S}$: $\mathbf{W}_{t+1} = \underset{\mathrm{vec}(\mathbf{W}) \in \mathcal{S}}{\mathrm{argmin}} \ \mathrm{Tr}\big((\mathbf{W} - \mathbf{Z}_{t+1})^\top (\mathbf{W} - \mathbf{Z}_{t+1})\big)$

End

Output: Final hypothesis $f(\mathbf{x})$: $H(\mathbf{x}, l) = \sum_{t=1}^{T} \eta_t \mathbf{1}\big(h_t(\mathbf{x}) = l\big)$
$f(\mathbf{x}) = \underset{l}{\mathrm{argmin}}\, H(\mathbf{x}, l)$

6.5 Lipreading Experiments with ISMA Features

For a comprehensive evaluation of the proposed method, several experiments were designed. Since our objective is to construct a speaker-independent lipreading system, all the experiments were carried out in the speaker-independent mode. The accuracies were then estimated by the one-speaker-out cross validation strategy. In all experiments, first, Mu-MABoost was trained to classify speech-units. The output of Mu-MABoost is a K-dimensional vector (K being the number of speech-units) where the i-th element is the probability of belonging to class i, given an input. This K-dimensional vector was computed for each frame and used as visual features with COBRA-selected ISCN features. The delta and delta-delta features were also included to construct the final visual feature set. These features were then used to train GMM-HMMs.

In AV-ASR, we also include the 39 MFCC features extracted from audio

signals in the feature set. Thirteen MFCC features and their first- and second-order derivatives were extracted from frames with 20 ms duration and 10 ms overlap. That is, the audio frame-rate was 100 frames per second. Visual features were linearly interpolated to increase their frame-rate from 30 to 100 to have the equal number of audio and visual features per utterance.

6.5.1 Oulu: Phrase recognition

The Oulu dataset consists of 10 daily used phrases such as *nice to meet you*. Each phrase is repeated 5 times by each of the 20 speakers.

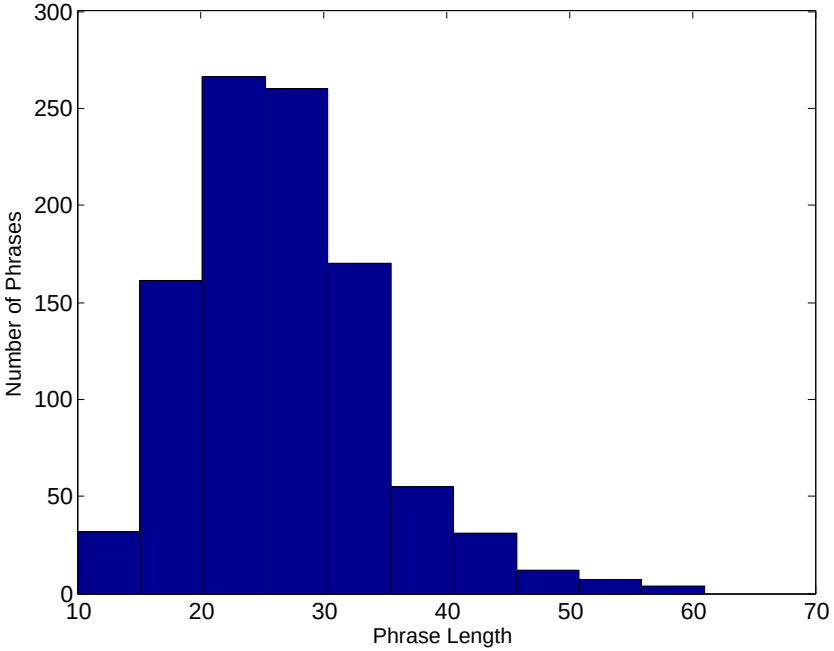

Figure 6.2: Histogram of length (number of frames) of phrases for the Oulu dataset. Most recordings are longer than 20 frames.

A brief description of this dataset can be found in Chapter 2. In each run,

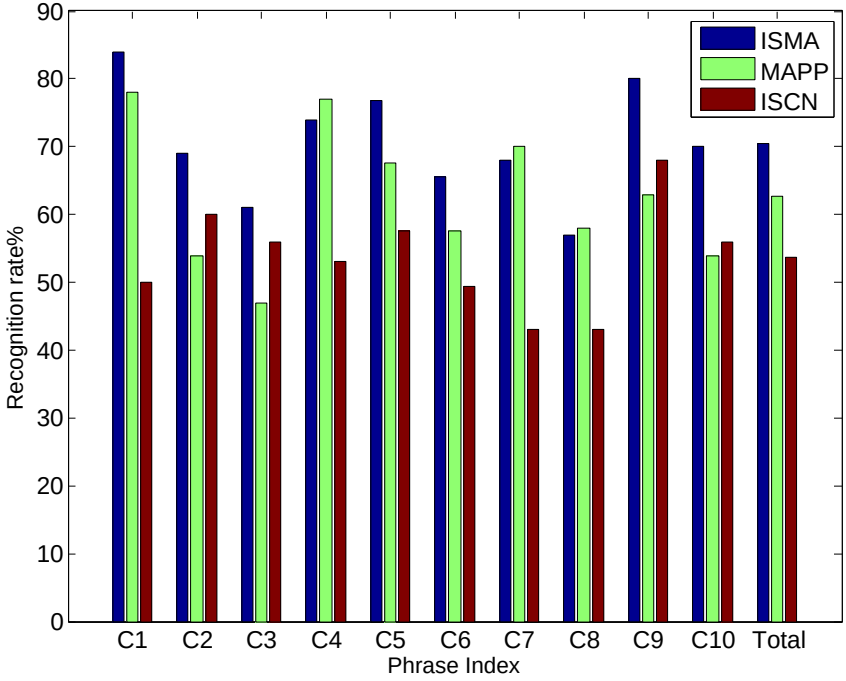

Figure 6.3: Phrase recognition accuracy for ISMA, MAPP and ISCN features. C1 to C10 are the ten phrases of the Oulu dataset listed in Table 2.3.

19 speakers were used to train HMMs. The main difference of this dataset from the GRID dataset was that phrases in the Oulu dataset are much longer than digits in the GRID dataset. The histogram in Figure 6.2 shows the length distribution of the Oulu phrases.

Each phrase was modeled by an HMM with 9 emitting states and GMMs with two Gaussian components and diagonal covariance matrices. The first set of experiments compares the performance of the Mu-MABoost features, i.e. when posterior probabilities are used to train GMM-HMMs, with ISCN features and reports the recognition rate when both feature sets were combined. In the following, the output posterior probabilities of Mu-MABoost

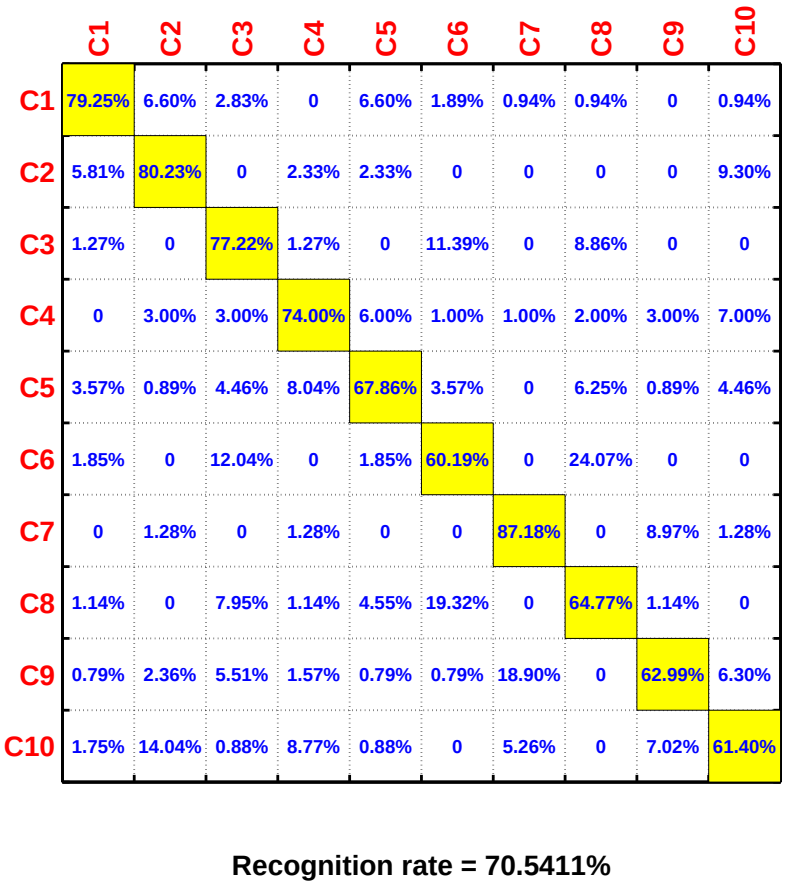

Figure 6.4: Confusion matrix of the visual recognizer with ISMA features. C8 (*thank you*) is the most wrongly classified phrase due to its confusion with C6 (*see you*). Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class.

were referred to as MAPP features and, the feature set containing both MuMABoost posterior probabilities and ISCN features was called ISMA in these experiments.

As seen in Figure 6.3, ISMA obtains 70.5% accuracy which is the highest

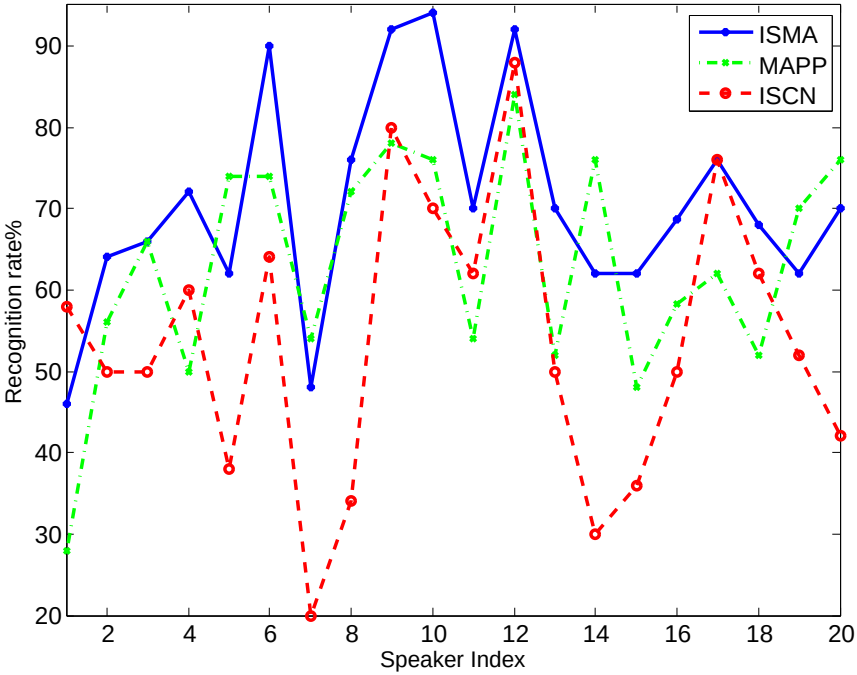

Figure 6.5: Recognition accuracy per person for speakers in the Oulu dataset. The ISMA accuracy varies from 45 to 94 percent over 20 speakers.

recognition rate reported for this dataset. MAPP and ISCN features obtain 63% and 55% accuracies, respectively. The worst phrase recognition rate is for *Thank you* (C8) due to its confusion with another similar phrase *see you* (C6). This confusion can be detected from the ISMA confusion matrix depicted in Figure 6.4. From the visual speech viewpoint, the only difference between these two phrases is in the tongue position which is usually not captured in videos. Grouping these two phrases into a one class, improves the final classification rate by 3.6%.

Figure 6.5 reports the recognition accuracy for each speaker. Although,

the inter-speaker variation of accuracies obtained by ISMA and MAPP features is smaller than ISCN features (standard deviation of ISMA accuracy over 20 speakers is 2.98 compared to 3.89 of ISCN features), it is still not sufficiently small to be called a reliable recognizer.

Feature sets	**ISMA+MFCC**	**MAPP+MFCC**	**ISCN+MFCC**	**MFCC**
∞ **dB**	88.27	88.28	79.36	94.99
10 **dB**	80.05	73.13	65.93	42.77
5 **dB**	71.84	61.74	58.52	20.93
0 **dB**	65.14	56.62	49.52	14.93
-5 **dB**	57.54	45.1	41.17	12.12
-10 **dB**	60.74	40.61	35.59	12.03
-15 **dB**	55.03	43.57	34.38	11.72
-20 **dB**	50.72	38.97	31.87	10.22

Table 6.1: Recognition rates in percentage at different noise levels for the audio-visual recognizer trained with 3 different feature sets and the audio-only recognizer trained with MFCC features. Second column: ISMA features with MFCC features extracted from audio data. Third column: Audio-visual recognizer trained with a feature set containing MAPP and MFCC features. Fourth column: ISCN with MFCC features and finally the fifth column reports the accuracy of the audio-only recognizer at various SNRs.

Table 6.1 reports the accuracy of audio-visual speech recognizer for ISMA with MFCC, MAPP with MFCC and ISCN with MFCC features when audio signals are corrupted by additive white noise at various SNR levels $\{-20, -15, -10, -5, 0, 5, 10, 15, \infty\}$. Accuracy of the audio-only recognizer is listed in the last column. Two interesting points can be detected in this table.

First, the performance of the audio-visual recognizer is worse than that of the audio-only recognizer in noise-free scenario. This is an important indication of unreliability of visual features. If visual features were reliable (that is, there was no mismatch between the distributions of visual features in the training and test sets), adding visual features to the audio feature set would only improve the performance, due to the Bayes theorem. However, as we observe, in reality adding a visual feature set to audio features may in fact reduce the performance due the speaker-dependent nature of visual cues.

Second, using visual cues in low-SNRs largely improves the recognition

accuracy. As seen, at 0 dB for instance, the audio-only recognizer performance is barely better than random, while the audio-visual recognizer can correctly classify 65% of the phrases.

6.5.2 GRID: Digit recognition

The second set of experiments were conducted on the GRID dataset [CBCS06]. GRID is a relatively large dataset with 32 speakers and 1000 recordings per speaker (for more details look at Chapter 2). In the following tests, we only used the recordings of the first sixteen speakers of this dataset. For each speaker, we extract digits 0-9 from the first 400 utterances. Since in

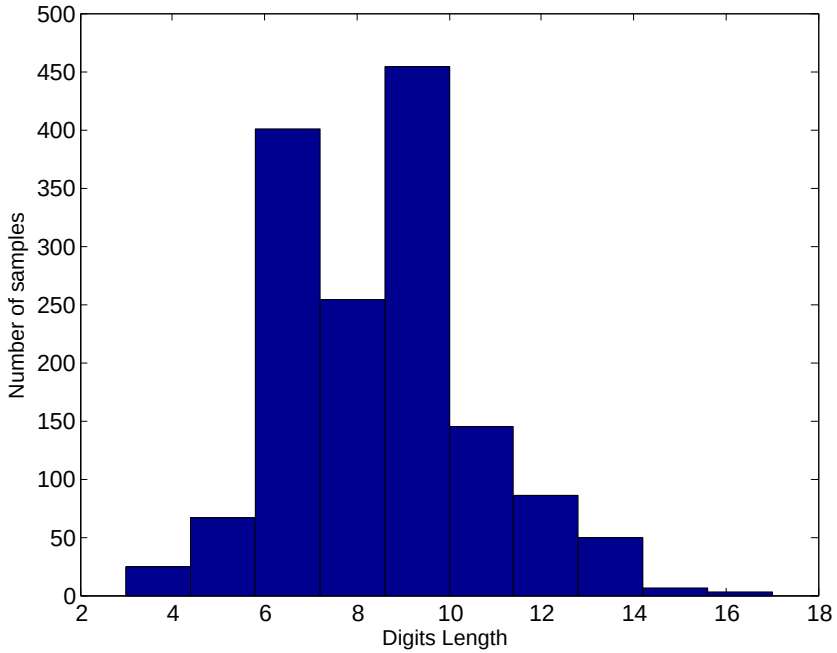

Figure 6.6: Histogram of digit lengths in the GRID dataset. Most digits were between 6 to 10 frames long.

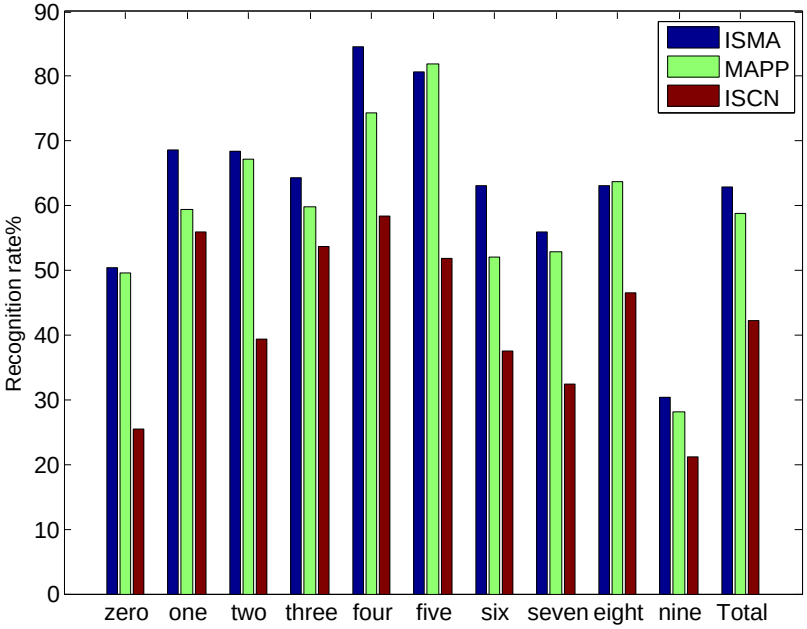

Figure 6.7: Digit recognition accuracy for ISMA, MAPP and ISCN features.

the GRID dataset talkers had 3 seconds to produce each sentence , the length of the spoken digits extracted from these sentences were much shorter than the length of phrases in the Oulu dataset. Moreover, for reducing the sample size, video data were down-sampled to 15 frames per second which further reduced the number of frames per digit. The histogram in Figure 6.6 shows the length distribution of the GRID digits.

As seen in Figure 6.6, most digits are shorter than 10 frames and many of them are as short as 6 frames. Hence, unlike the Oulu dataset, we use HMMs with 3 emitting states to model the GRID digits. As in the previous experiments, we first compare the performance of the MAPP features, i.e. when posterior probabilities are used to train GMM-HMMs, ISCN features and ISMA features which are the combination of MAPP and ISCN features.

Figure 6.7 compares the classification power of ISMA, MAPP and ISCN

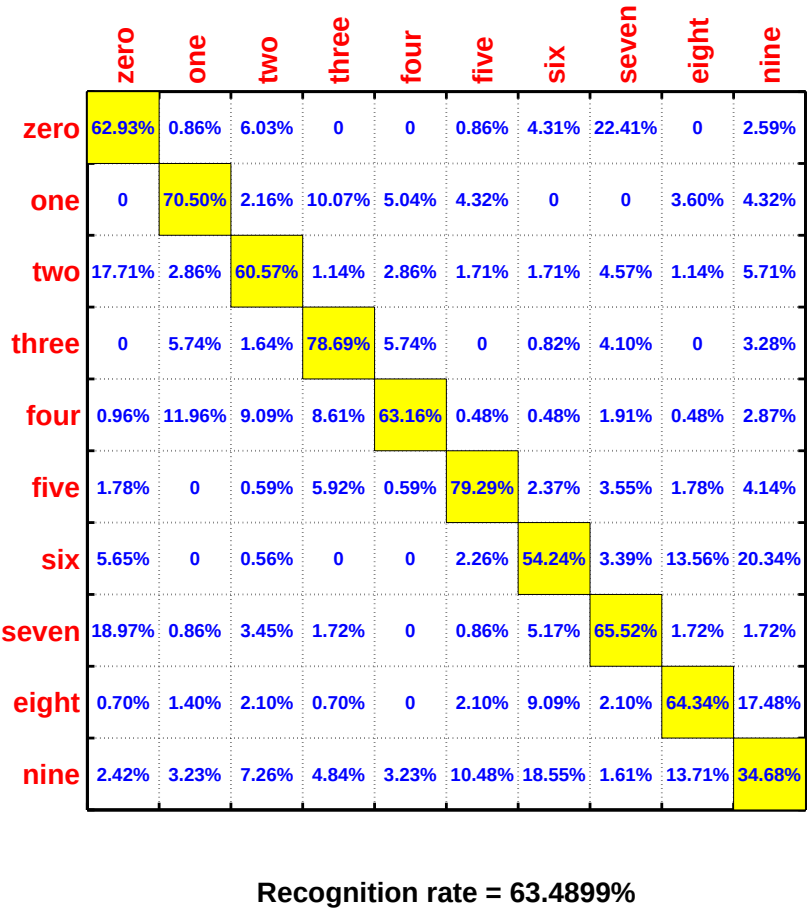

Figure 6.8: GRID dataset: Confusion matrix of the visual recognizer with ISMA features. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class.

features. Similar to the Oulu dataset, ISMA features obtain the highest accuracy, i.e., 63%, followed by the MAPP features with 59.5% digit classification rate. Moreover, 50% of occurrences of all digits except *nine* are correctly classified by ISMA features.

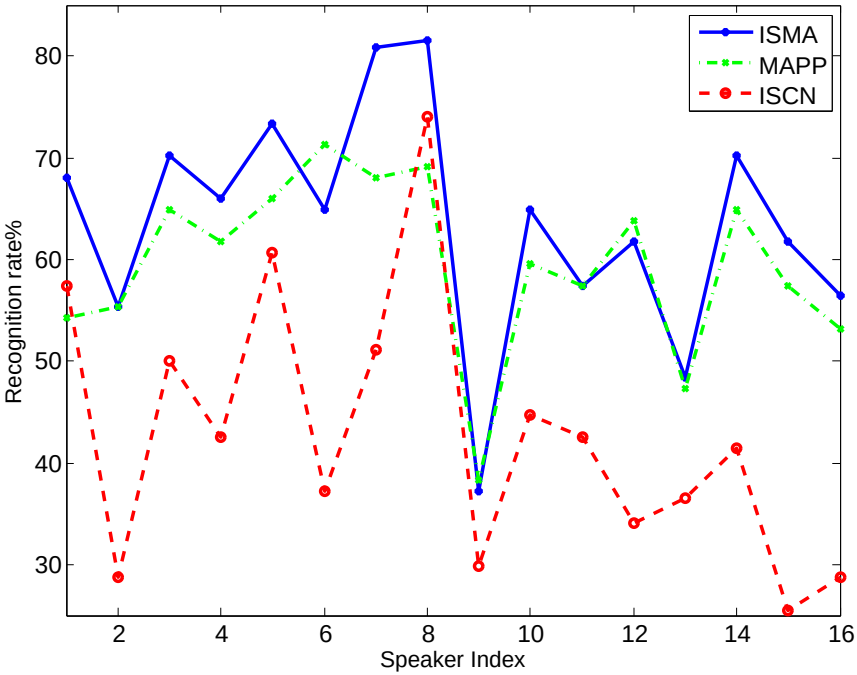

Figure 6.9: Recognition accuracy per person for the first 16 speakers of the GRID dataset. The ISMA accuracy varies from 37 to 81 percent over 16 speakers. Removing the ninth speaker from the dataset will 2% increase the mean accuracy.

The confusion matrix depicted in Figure 6.8 reveals that *nine* is commonly confused with *eight* or *six*. This confusion can be attributed to the fact that their vowels, i.e., /ay/, /ey/ and /i/ are all mapped to the same viseme /I (see Table 3.10). While ISMA obtains promising accuracy of 63% on such a difficult dataset (due to the short length of digits), its variation over the speakers is still not satisfactory. As seen in Figure 6.9, the ISMA accuracy is as low as 37% on one speaker. This is due to the fact that lipreading is by nature speaker-dependent. It is a common observation that some people hardly move their lips while speaking. This is translated to different degrees

of visemes visibility for different speakers, due to their lip shapes, facial characteristics and speaking styles.

Feature sets	**ISMA+MFCC**	**MAPP+MFCC**	**ISCN+MFCC**	**MFCC**
∞ **dB**	81.29	87.66	72.02	95.48
10 **dB**	73.13	74.6	58.97	45.88
5 **dB**	70.32	69.51	54.39	29.19
0 **dB**	67.04	63.28	49.64	17.69
-5 **dB**	64.41	56.64	44.4	11.3
-10 **dB**	62.01	51.41	39.63	11.24
-15 **dB**	60.53	48.06	37.01	11.17
-20 **dB**	58.93	46.66	35.6	11.24

Table 6.2: Recognition rates in percentage at different noise levels for audio-visual recognizer trained with 3 different feature sets and audio-only recognizer trained with MFCC features. Second column: ISMA features with MFCC features extracted from audio data. Third column: Audio-visual recognizer trained with a feature set containing MAPP and MFCC features. Fourth column: ISCN with MFCC features and finally the last column reports the accuracy of the audio-only recognizer at various signal-to-noise ratios.

Table 6.2 reports the accuracy of the audio-visual digit recognizer when (I) ISMA as visual features and MFCC as audio features are used, (II) MAPP and MFCC features are employed and (III) ISCN and MFCC are used to train the recognizer. The last column reports the accuracy of the audio-only recognizer at various signal to noise ratios. A surprising result is that ISMA+MFCC features obtains significantly worse results than MAPP+MFCC features on clean speech. However, as the SNR decreases, the ISMA+MFCC tends to outperform the MAPP+MFCC features as expected. This is due to the fact that the number of ISMA features (100) is 5 times larger than the MAPP features and when used with 39 MFCC features, they dominate the value of the loglikelihood. With a proper weighting scheme this effect can be substantially alleviated. As in the Oulu dataset, we again see that the performance of the audio-only recognizer can be largely improved at low-SNRs. At high-SNR values however, the audio-only recognizer still works better than the audio-visual recognizer, meaning that a more complex fusion scheme is needed to properly weight audio and visual modalities according to their contributions.

6.5.3 AVletter: Letter recognition

In this section a letter recognizer was trained to classify 26 English letters A-Z. Each English letter, except W, can be transcribed by a sequence of 1 to 3 phonemes. Since visual speech has much lower time-resolution than audio speech, a speech unit as small as a phoneme may only be captured in one frame, which is hardly enough to learn its corresponding statistics. Modeling longer speech-units (but not as long as words), such as a short sequence of phonemes as in this dataset, allows us to more precisely model visual speech while it is still feasible to generalize it to a large-vocabulary automatic speech recognition task.

Figure 6.10 compares the recognition accuracy of ISMA, MAPP and ISCN features. Unlike the previous datasets, ISCN and MAPP features achieve close accuracy rates (32% and 33%, respectively) on AVletter. ISMA however, reaches 40.3% accuracy rate. While on some letters such as Y, W,

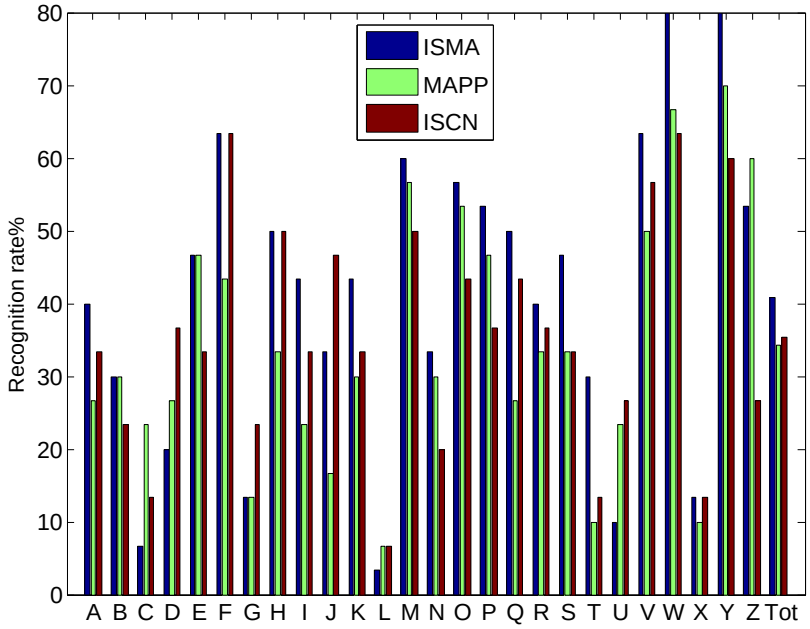

Figure 6.10: Recognition accuracy of letters for ISMA, MAPP and ISCN features.

M and F the accuracy rate is over 70%, for some letters, the visual recognizer could hardly achieve a performance better than random guessing. In the AVletter dataset, each letter is repeated 30 times by 10 different speakers. The confusion matrix of the letter recognition task is quite informative. As shown in Figure 6.11, some letters such as C, D and T which have exactly the same visual transcription, i.e. they consist of exactly the same sequence of visemes, are commonly confused. For some other letters such as U and Q which have different viseme transcriptions, the source of the confusion is the fact that they have the same dominant viseme. For instance, the dominant viseme of U and Q is /B (which corresponnds to phoneme /uh/; see Table 3.10). By governing the lip shape and prevents other visemes from having visible effects on the visual speech, /B hides other visemes and results in confusion of Q and U. The same argument holds for the confusion of S and X (in this case the dominant viseme is /I which corresponnds to phoneme /s/).

In lipreading visemes are considered the smallest visibly distinguishable speech units. However, since there is no unique mapping from phonemes to visemes, different mappings result in different viseme sets. From the classification viewpoint, the best viseme group is a set of visemes that are highly separable given the visual features. Having this definition in mind, it is interesting to investigate whether the commonly accepted viseme set listed in Table 3.10 is optimal. In order to find a data-driven optimal viseme set with k visemes that minimize the classification error, we grouped the 26 English letters into 13 classes by partitioning the confusion matrix into 13 almost disjoint principal submatrices, i.e., the rows and columns should be re-arranged in order to obtain a new matrix where 13 non-zero principal submatrices are observable. While the off-diagonal elements of these principal submatrices

Visual Groups		
{I, R}	{S, X}	{M, N}
{J, Z}	{B, P}	{O}
{F}	{W}	{A, E, K, L}
{H}	{G, Y}	
{C, D, T, V}	{Q, U}	

Table 6.3: Regrouping 26 English letters to a smaller set containing 13 visibly distinguishable elements.

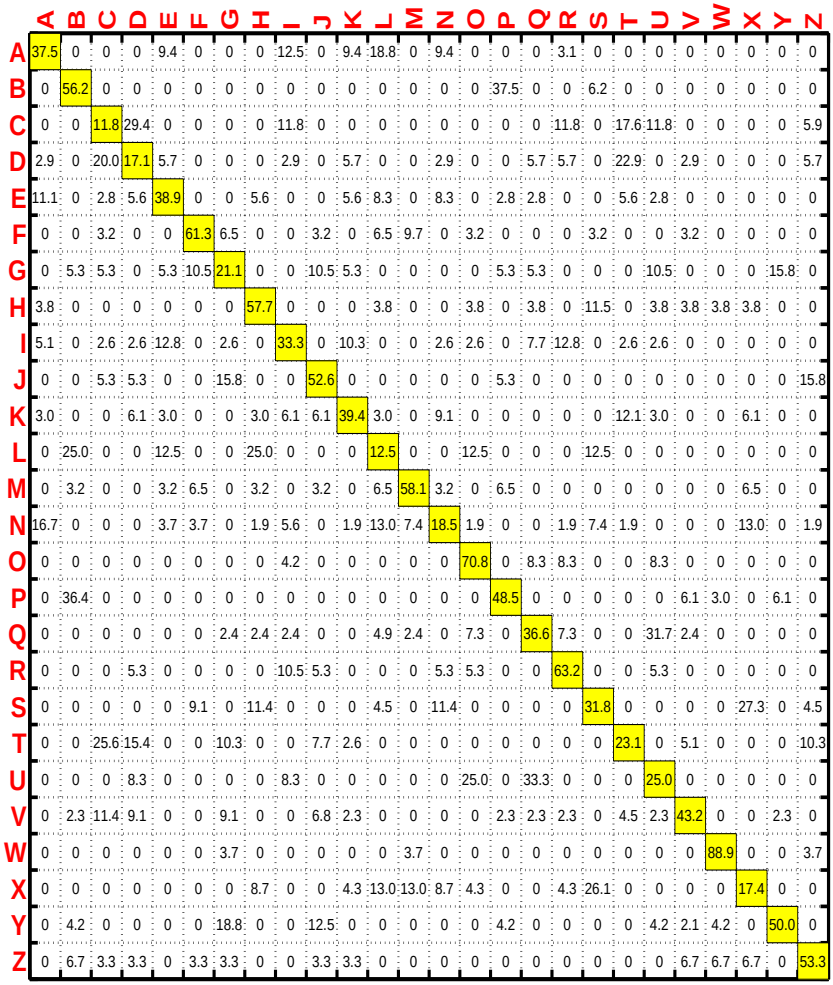

Figure 6.11: AVletter dataset: Confusion matrix of Visual recognizer with ISMA features.

are non-zero, the off-diagonal elements of inter-principal submatrices should be small. This problem however, is NP-hard. A commonly used approximation to find a feasible solution for this problem is to use the spectral clustering method [vL07]. Applying this method to the confusion matrix given in Figure 6.11, yields the mapping described in Table 6.3. Using these 13 classes, instead of 26 English letters, increases the classification rate to 61.3%. As can be seen in Table 6.3, most of the letters with similar visemic transcriptions such as C, D and T are grouped together. S and X are also put in the same class and Q and U are also considered indistinguishable.

6.6 Conclusion and Discussion

In this chapter, we developed a lipreading system based on the multiclass MA-Boost algorithm presented in Chapter 4 and COBRA-selected ISCN features introduced in Chapter 3. We showed that, training a multiclass classifier with 3D-SIFT features extracted from multiple color spaces and using the output posterior probabilities as visual features with ISCN features yielded a promising lipreading system. This method obtains 70.5% accuracy rate on the Oulu dataset which is 6% higher than spatiotemporal local binary patterns (LBP) based method reported in [ZBP09], and yields 63.5% digit recognition accuracy on the GRID dataset which is 4% higher than the AAM based algorithm suggested in [LHT$^+$09]. Moreover, both methods described in [ZBP09] and [LHT$^+$09] have employed some semi-automatic lip-region detection method which clearly improves the final accuracy, while in our work we only used a fully automatic facial detection point method described in [DGFVG12].

ISMA features, which are the combination of MAPP and ISCN features, consistently outperform other methods over 3 audio-visual datasets that were employed in this Chapter. However, the inter-speaker variation of the accuracy of the ISMA features is still not sufficiently small. While for most speakers, its accuracy varies in an acceptable range, in each dataset, there are some speakers that their corresponding lipreading recognition accuracies were significantly lower than the average accuracy due to their particular speaking styles, lip shapes and other facial characteristics. As shown in our experiments, the mismatch between training and test conditions leads to an audio-visual recognizer with worse performance than that of the audio-only recognizer when speech signals are clean. However, as the SNR values of the audio

signals decrease, the audio-visual recognizer outperforms audio-only recognizer with a large margin. In order to obtain an audio-visual recognizer that is favorably comparable with the audio-only recognizer in noise-free scenarios and outperforms it in noisy environments, we need a more sophisticated information fusion algorithm that weights the audio and visual modalities according to their information values. Such a method is developed in the next Chapter.

Chapter 7

Audio Visual Information Fusion

Information fusion in multi-sensory systems is a challenging task due to the fact that different sensors capture underlying phenomena differently. This raises many problems such as having data streams with different temporal rates, various data representations, different dynamic ranges, different sensitivity levels and different noise types.

This chapter looks into the multi-modal information fusion problem in the context of audio-visual speech recognition. By investigating the sources of uncertainty in the system, i.e., the estimation and model error, it is shown that a more suitable criterion to estimate the reliability weights assigned to modalities is to maximize the area under a receiver operating characteristic curve (AUC) rather than existing criteria, e.g., recognition accuracy or mutual information. Moreover, here we estimate a reliability weight for each feature. This generalizes the (conventional) two dimensional stream weight estimation problem to a fairly high dimensional problem. In order to efficiently estimate the reliability weights, we use a smoothed AUC function and adopt a variant of the projected gradient descent algorithm to maximize the AUC criterion in an online manner.

The proposed algorithm results in a robust information fusion scheme that can cope with audio-modality failure. Moreover, it is shown that by using the one weight per feature strategy the audio recognition rate is greatly improved

in the noisy environments.

7.1 The Problem of Audio Visual Fusion

Having multiple observation streams in a pattern recognition task, the central question is how to optimally combine their information. Various information fusion algorithms suggested in the context of audio-visual automatic speech recognition (AV-ASR), can be categorized into two groups. First, feature fusion methods where audio and video features are concatenated to constitute a super-feature vector. Linear discriminant analysis (LDA) or other dimension-reduction methods are usually followed to reduce the final feature vector dimensionality. Though straightforward from a mathematical viewpoint, these approaches give the impression of mixing the unmixable and have been often reported to be inferior to the decision fusion techniques [PNLM04]. The second group of approaches, i.e., decision fusion methods, analyze the different streams separately and combine the decisions of the single modality classifiers at the state level or even at the phoneme or word level in HMM-based recognizers. This strategy allows to optimize each classifier for the characteristic of its input features and thus achieves higher accuracy.

A common decision fusion method of combining information sources, or as we call here *feature streams*, in a statistical pattern recognition framework is to use the product rule on their posterior probabilities, i.e., Bayes fusion [KHDM98]. This approach results in a naive Bayes classifier which coincides with the Bayes classifier when the feature streams are class conditionally independent. Despite the theoretical appeal of Bayes fusion, in reality however, naive Bayes classification may severely deviate from Bayes classification due to the following reasons.

1. **Model error**: Any assumption of a parametric distribution may cause a mismatch between the *true* and the *estimated* distribution [RCB97].

2. **Estimation error**: Parameter estimation error which mainly originates from insufficient training data may shift the decision boundary and thus, impose an additional uncertainty. Moreover, the mismatch between the distributions of training and test data (because of the time-varying environment) also can be seen as a result of insufficient training data.

3. **Independence Assumption**: The dependency among features violates the central assumption in naive Bayes classification.

Different approaches have been suggested to alleviate these problems. For instance, in [PLHZ04] audio and visual streams are coupled through a low dimensional random variable (in fact two dimensional in their work). That is, the class conditional distribution consists of multiplication of three terms: the likelihood of audio stream, the likelihood of video stream and a factor that models their dependency. This interesting idea, however, is only capable of dealing with the violation of the independence assumption.

A more commonly used approach to tackle the above problems is through assigning a reliability weight (an exponent weight) to each stream pdf. This approach has been empirically shown to be effective [WSC99], [CR98], [CDM02], [DL00] in audio-visual speech and speaker recognition, especially in the presence of varying additive acoustic noise. Its effectiveness was later theoretically explored in [PSSD06] to derive an unsupervised weight estimation scheme. They assumed that the difference between the weighted likelihood and the Bayes likelihood (mismatch error) is a random variable with normal distribution and showed that a proper choice of weights can minimize the variance of the mismatch error. In fact, based on their approach it can be seen that any (linear or non linear) function of stream pdfs may be used to minimize the variance of the mismatch error. For instance, in [LCSC05] it was proposed to use a threshold-based combination of the sum rule and the product rule to fuse the class conditional likelihoods. It is, however, noteworthy that minimizing the variance of the mismatch error does not necessarily minimize the classification error. In fact, as shown by Friedman et al. [FF97], under some conditions the exact opposite may come true, i.e., larger variances result in lower classification error rates.

No matter what functional form is assumed to combine the stream pdfs (exponent weights, linear combination or etc.), we need a criterion to estimate the fusion parameters. Minimum word error rate is the most common criterion employed to estimate the parameters. In [PG98] for instance, a discriminative training method was proposed to minimize the smoothed error function (instead of the 0-1 loss function). Other existing criteria are maximum SNR value and maximum mutual information [ONS99] where the reliability of each stream is directly measured by mutual information between the stream and HMM states.

The main shortcoming of these criteria is that they are only optimum

for one realization of training data, i.e., the training data at hand. Paradoxically, however, these approaches are supposed to perform well in mismatching training and test conditions where the test data is corrupted with different background noises and thus clearly has a different distribution than the training data. Even more importantly, it is a well-known fact that visual features are highly speaker-dependent [CHLN08]. That is, the distribution of visual features in training data, may vary from those in test data. A justification for this rather paradoxical parameter estimation procedure is that in the absence of information about the test condition, the obvious strategy is to minimize the error rate over training data with the hope that it will generalize enough to achieve satisfactory results on the test data.

Adaptive weight estimation is a commonly suggested approach to tackle this problem [GVN$^+$01], [RS12]. This approach, however, may suffer from several issues including poor weight estimation accuracy due to the unsupervised nature of this approach, heavy computational complexity (since most of these approaches need to estimate the audio and video quality in real-time) and system complexity. The main question answered in this work is whether it is possible to estimate the fusion weights in a static, supervised manner and yet achieve a robust recognition system even in mismatch conditions.

The contribution of this work is threefold:

1. We suggest to use the area under a receiver operating characteristic curve (AUC) to estimate the exponent weights. AUC is linearly related to the expectation of classification accuracy and thus, the weights maximizing AUC achieve satisfactory results over a wider range of mismatch conditions.

2. We assign a reliability weight to each feature rather than each modality. This approach will particularly allow us to model the inter-dependency of visual features more effectively by increasing the hyper-parameters that describe the final combined pdfs.

3. In order to maximize the AUC value, a non-convex approximation of AUC is adopted and optimized by a gradient decent type algorithm.

AUC is a commonly used measure to evaluate the classification rules and ranking algorithms [HT01]. Interestingly, AUC can be seen as the average of the classification rates over the different classification thresholds and uniform distribution of classification costs (equivalently in our work, uniform

distribution over the priori probabilities of classes). Based on this interpretation of AUC, maximizing the AUC value is equivalent to minimizing the expectation of the classification rate. We relate the notion of classification threshold to mismatch error and show that the weights maximizing the AUC value (minimizing the average of error rate), result in more robust AV-ASR than the weights that are estimated to be optimum for only one realization of mismatch error.

Unlike the conventional methods where only two stream weights[1], one for audio and one for video, are used to integrate the audio-visual information sources, in this work a reliability weight is assigned to each feature. This generalization is usually referred to as the weighted naive Bayes classifier in the literature. There is a rich body of theoretical and empirical work on the weighted naive Bayes classifier [ZCCW13, and references therein]. It is also a fairly common approach in audio only ASR, though, to the best of our knowledge it has not been utilized for AV-ASR. Moreover, we further generalize this model by assigning an individual weight vector to each class.

For at least two reasons, these generalizations are beneficial: First, these weights are effective tools to model the relatively high correlation among visual features. Second, different visual (audio) features may have different Bayes error (informativeness) and different level of robustness against mismatch error.

However, due to the increase in the number of parameters, a computationally efficient method is required to estimate the weights. To this end, we use a variant of the online projected gradient descent algorithm [Zin03] that as shown in [ZJYH11] can efficiently maximize the AUC value over a large dataset.

7.2 Preliminaries & Model Description

Throughout this work, we consider an instance of the pattern classification problem. Given a training dataset $\mathcal{D} = \{(\mathbf{O}_i, y_i)\}, 1 \leq i \leq N$, containing audio-visual recordings (samples) $\mathbf{O}_i$ of K distinct words (classes), i.e., $y_i \in \{1, \ldots, K\}$, the goal is to train a classifier that optimally assign a class label y to any new test sample. Following the common approach in AV-ASR,

[1] In fact, in most of the previous works the summation of the weights is constrained to be one and consequently, only one weight needs to be estimated.

a multi-stream HMM is trained for each class to capture its statistical properties and represent the class conditional distribution. The models and class conditional distributions are denoted by M_k, $1 \leq k \leq K$ and $P(\mathbf{O}|M_k)$, respectively. The Bayes classification rule can be used then to assign a class label to any given sample $\mathbf{O}$:

$$\hat{y} = \arg\max_{1 \leq k \leq K} P(\mathbf{O}|M_k)\pi_k = \arg\max_{1 \leq k \leq K} \log P(\mathbf{O}|M_k) + \log \pi_k \tag{7.1}$$

where $\pi_k = P(y = k)$ is the a priori probability of class k.

A common practice in speech recognition, including this work, is to approximate $P(\mathbf{O}|M)$ with the probability of the most probable state sequence of M computed by the Viterbi algorithm.[2]

Given a sample $\mathbf{O} = \{\mathbf{o}_{AV}^1, \ldots, \mathbf{o}_{AV}^T\}$ (where $\mathbf{o}_{AV}^t = [\mathbf{o}_A^t, \mathbf{o}_V^t]$ is a bimodal observation vector for this sample at time t), the state emission probability (observation probability) of an HMM at state S at time t can be written as:

$$P(\mathbf{o}_{AV}^t|q_t = S) = P(\mathbf{o}_A^t|q_t = S)^{w_A} P(\mathbf{o}_V^t|q_t = S)^{w_V} \tag{7.2}$$

where w_A and w_V are appropriate stream weights to account for the reliability difference of each modality and $\mathbf{o}_A^t = [o_{A_1}^t \ldots, o_{A_{|A|}}^t]$ and $\mathbf{o}_V^t = [o_{V_1}^t, \ldots, o_{V_{|V|}}^t]$ are audio and visual observation vectors, respectively. Moreover, $|A|$ and $|V|$ are the dimensionalities of audio and visual feature vectors. This is a fairly standard multi-stream likelihood model. Assuming independence among audio features and among visual features, we further generalize this model by adopting reliability weight for each feature. That is,

$$P(\mathbf{o}_{AV}^t|q_t = S) = \prod_{i=1}^{i=|A|} P(o_{A_i}^t|q_t = S)^{w_{A_i}} \prod_{j=1}^{j=|V|} P(o_{V_j}^t|q_t = S)^{w_{V_j}} \tag{7.3}$$

Suppose $\mathcal{Q} = (q_1, ..., q_T)$ is a sequence of random variables taking values in some finite set $\mathcal{S} = \{S_1, ..., S_n\}$, the state space. Given a model M and an input sample $\mathbf{O} = \{\mathbf{o}_{AV}^1, \ldots, \mathbf{o}_{AV}^T\}$, the log-likelihood can be approximated

[2] Since we use the weighted pdfs, this approximation does not give rise to a valid probability distribution function anymore, though, for simplicity we still refer to it as likelihood.

as follows:

$$\log P(\mathbf{O}|M) \approx \log \hat{P}(\mathbf{O}|M) = \max_{\mathcal{Q}} \log P(\mathbf{O}, \mathcal{Q}|M) \tag{7.4}$$
$$= \log P(\mathbf{O}, \mathcal{Q}_*|M) = \sum_{t=1}^{T} \log P(\mathbf{o}_{AV}^t|q_t) + \sum_{t=1}^{T} \log a_{q_{t-1},q_t}$$

where $\mathcal{Q}_* = (q_1, \ldots, q_T)$ is the most probable sequence of states in M computed by the Viterbi algorithm, $\hat{P}(\mathbf{O}|M)$ is the approximation of $P(\mathbf{O}|M)$ and a_{q_{t-1},q_t} are the transition probabilities with a_{q_0,q_1} equal to 1. By substituting $P(\mathbf{o}_{AV}^t|q_t)$ from (7.3) into (7.4), the (approximation of) log-likelihood can be written as a linear function of the reliability weights:

$$\log \hat{P}(\mathbf{O}|M) = \sum_{t=1}^{T} \sum_{s \in \{A,V\}} \sum_{j=1}^{|s|} w_{s_j} \log P(o_{s_j}^t|q_t)$$
$$+ \sum_{t=1}^{T} \log a_{q_{t-1},q_t} = \mathbf{w}^\top \mathbf{p} \tag{7.5}$$

where $\top$ stands for the vector transpose and,

$$\mathbf{w} = [w_0, w_{A_1}, \ldots, w_{A_{|A|}}, w_{V_1}, \ldots, w_{V_{|V|}}]^\top \tag{7.6}$$
$$\mathbf{p} = \sum_{t=1}^{T} [\log a_{q_{t-1},q_t}, \log P(o_{A_1}^t|q_t), \ldots, \log P(o_{V_{|V|}}^t|q_t)]^\top$$

Each entry $p[l]$, $l \in \{1, ..., |A| + |V|\}$ of $\mathbf{p}$, indicates the amount of contribution of that feature to the total log-likelihood value. The first entry $p[0]$ is the contribution of transition probabilities to the loglikelihood and w_0 is a weight to adjust this term. w_0 can either be fixed to 1 or can be considered as a variable to be optimized.

The next step to estimate the reliability weights is to select an appropriate criterion. Let denote the criterion by $f(\log P(\mathbf{O}|M))$ which is a function of log-likelihood, the goal is to optimize f with respect to $\mathbf{w}$. It is important to remember that given an optimal state sequence $\mathcal{S}_*$, log-likelihood is a linear function of $\mathbf{w}$. However, in order to find the optimal state sequence $\mathcal{S}_*$ itself, the Viterbi algorithm needs $\mathbf{w}$. Thus, the optimization process $\max_{\mathbf{w}} f(\log P(\mathbf{O}|M))$ can be approximately solved by the following iterative process which jointly optimizes $\mathbf{w}$ and $\mathcal{S}$.

Algorithm 7.1: Iterative joint optimization

Input: set $\mathbf{w}(1) = [1, \ldots, 1]^\top$, $\epsilon = \infty$ and θ =stopping threshold
while $\epsilon > \theta$ **do**

$$\mathcal{Q}_j = \max_{\mathcal{Q}} \log P(\mathbf{O}, \mathcal{Q}|M, \mathbf{w}(j)) \tag{7.7}$$

Calculate $\mathbf{p}_j$ from (7.6) (7.8)

$$\mathbf{w}(j+1) = \arg\max_{\mathbf{w}} f(\mathbf{w}^\top \mathbf{p}_j) \tag{7.9}$$

$$\epsilon = ||\mathbf{w}(j+1) - \mathbf{w}(j)||_2$$

$$j++.$$

end
Output: The final weight vector $\mathbf{w}$.

The $\mathbf{w}(j)$ argument in (7.7) is to remind that Viterbi algorithm needs the values of the weights in order to compute the optimal path and θ is a stopping threshold usually set to a small value. In our work it was set to 10^{-6} and at most three iterations were sufficient to converge.

Throughout the rest of the chapter we mainly focus on the criterion function f and how to optimize it, i.e., equation (7.9) of Algorithm 1.

Classification accuracy is a commonly used criterion in order to to estimate the free parameters of a model in a classification task. However, as we discussed before, this criterion results in a sub-optimal solution in mismatched training-testing scenarios. Instead, we use a more robust criterion that takes the mismatch error into account, that is, maximizing the AUC value.

7.3 AUC: The Area Under an ROC Curve

Receiver operating characteristic (ROC) curves are commonly used two-dimensional graphs which are originally developed in classical detection theory and later found their applications in machine learning due to their invariance to monotonic score transformation and class distributions. An ROC curve is defined as a plot of true positive (tp) rate on the Y axis against false positive (fp) rate on the X axis and thus it is intrinsically defined for binary

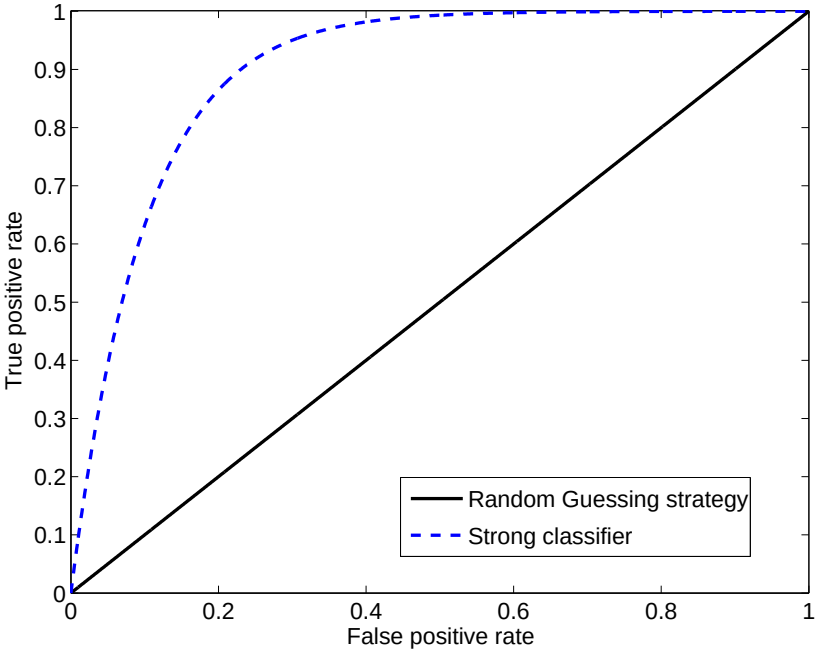

Figure 7.1: An example of receiver operating characteristic curve.

classification (detection) problems.

$$\text{tp} = \frac{\text{Positives correctly classified}}{\text{Total positives}}$$
$$\text{fp} = \frac{\text{Negatives incorrectly classified}}{\text{Total negatives}}$$

From its definition, it is clear that the points lying on the $Y = X$ line belong to random guessing strategy and the point $(0, 1)$ represents optimal classification (look at Figure 7.1). Given a confidence-rated classifier[3] a threshold (or in general decision boundary) can be used to produce the final binary decision.

[3]Confidence-rated classifiers yield an instance probability or score, a numeric value that represents the degree to which an instance is a member of a class.

Each threshold value produces a different point in ROC space. Clearly, when threshold is $-\infty$ both tp and fp are 0 and as threshold approaches ∞ both of these values are 1. For any threshold value in between, we get a point on a concave curve above the $Y = X$ line as in Figure 7.1. AUC is then simply the area under the ROC curve.

Following the AUC interpretation suggested in [CM03], consider a binary classification task with m positive samples and n negative samples. A confidence-rated classifier assigns a score to each sample. We assume the classifier outputs for positive samples $x_1^+, \ldots, x_m^+$ and negative examples $x_1^-, \ldots, x_n^-$ are ordered. The AUC is then calculated as:

$$A = \frac{1}{mn} \sum_{i=1}^{i=m} \sum_{j=1}^{j=n} \mathbb{I}_{(x_i^+ > x_j^-)} \tag{7.10}$$

where $\mathbb{I}_{(.)}$ is one if $(.)$ holds, and zero otherwise. We remark that the AUC value is exactly the probability $P(X^+ > X^-)$, where X^+ is the random variable corresponding to the distribution of the outputs for the positive samples and X^- the one corresponding to the negative samples. This interpretation was in fact the inspiration to use the AUC in this work. Under this interpretation, AUC can be seen as a measure of the ranking quality of the classifier. As the AUC value increases, it becomes more probable that the score of a positive sample is larger than the score of any negative sample in the dataset. That is, in an audio-visual dataset, the scores of positive samples are likely to be higher than negative samples, independent of their speakers and the mismatch error corresponding to samples.

AUC, however, can be seen from another point of view, that is as the expectation of the classification accuracy. Consider a binary classification problem with two HMMs M_1 and M_2 trained with samples from the first and second class, respectively. We distinguish the "error free" class conditional probabilities (free from estimation error and not from modeling error) by B subscription, i.e., $P_B(\mathbf{O}|M_k)$, from estimated likelihoods. Since we need a confidence-rated classifier, we have to define a discriminant function as follows to assign a score to each sample.

$$L(\mathbf{O}) = \log P_B(\mathbf{O}|M_1) - \log P_B(\mathbf{O}|M_2) + \log(\frac{\pi_1}{\pi_2}) \tag{7.11}$$

From (7.1) it is clear that based on the optimal classification rule, $\mathbf{O}$ belongs to class 1 if $L(\mathbf{O}) \geq t$, and to class 2 otherwise and t is the classification

threshold which is equal to zero in the Bayes rule. Let $g_1(L) = P(L(\mathbf{O})|y = 1)$ be the probability distribution function of the scores of the samples belonging to class 1 and $g_2(L)$ be the pdf of the scores of class 2. The classification error rate, then, can be written as:

$$e = c\pi_1 \int_{-\infty}^{t} g_1(L)dL + (2-c)\pi_2 \int_{t}^{+\infty} g_2(L)dL \tag{7.12}$$

where $0 \leq c \leq 2$ is the classification cost. Consider both classification cost and threshold as random variables. Based on the following lemma according to Flach et al. [FHF11], AUC can be seen as a linear function of the expectation of classification error e where expectation is taken with respect to c and t.

Lemma. *Expected error for uniform cost distribution and uniform instance selection decreases linearly with AUC as follows*:

$$\mathbb{E}_{c,t}\{e\} = 2\pi_1\pi_2(1 - \text{AUC}) + \frac{\pi_1^2 + \pi_2^2}{2} \tag{7.13}$$

where $\mathbb{E}_x$ is the expectation with respect to to the random variable x. Thus, the AUC is in fact the average of classification rates over different classification thresholds and different classification costs. In the next subsection we connect the notion of classification threshold and classification cost to mismatch error and show that according to the above lemma, AUC can be interpreted as the expectation of the classification error over different mismatch conditions.

7.3.1 AUC and Mismatch

Let us denote the difference between the true discriminant function and the estimated discriminant function by $Z(\mathbf{O})$, i.e.,

$$L(\mathbf{O}) - \hat{L}(\mathbf{O}) = Z(\mathbf{O}) \tag{7.14}$$

where $\hat{L}$ is the estimated discriminant function. In the absence of a priori knowledge about mismatch error, the classification decision can be made by $\hat{L}$, that is, a sample $\mathbf{O}$ belongs to class 1 if $\hat{L}(\mathbf{O}) \geq 0$, and to class 2 otherwise. However, this classification rule is equivalent to $L(\mathbf{O}) \geq Z(\mathbf{O})$. If we further assume that mismatch error $Z(\mathbf{O}) = Z$ is constant over all the samples in a given dataset, then Z can be seen as the classification threshold. In other words, mismatch error results in shifting the classification threshold from zero

to Z. Note that, Z is a random variable varying from one dataset to another. Denote the mismatch error of the training data and the test data by Z_{tr} and Z_{te}, respectively. Reliability weights are usually selected in order to minimize the classification error over the training data which is given by:

$$\begin{aligned} e_{tr} &= \pi_1 \int_{-\infty}^{0} g_1(\hat{L}(\mathbf{w}))d\hat{L} + \pi_2 \int_{0}^{+\infty} g_2(\hat{L}(\mathbf{w}))d\hat{L} \\ &= \pi_1 \int_{-\infty}^{Z_{tr}} g_1(L)dL + \pi_2 \int_{Z_{tr}}^{+\infty} g_2(L)dL \end{aligned} \tag{7.15}$$

where the argument $\mathbf{w}$ is to emphasize that the estimated score $\hat{L}(\mathbf{w})$ depends on the weight vector $\mathbf{w}$. However, since Z_{tr} is likely to be different from Z_{te}, the estimated weights are not optimal for the test condition. Moreover, a priori probabilities of the test data may significantly vary from the training data. This difference can be modeled by classification cost c in (7.13). Thus, a more reasonable strategy to select the reliability weights is to minimize $\mathbb{E}_{c,Z}\{e\}$ where the expectation over c is for averaging over different class priors and the expectation over Z accounts for different levels of mismatch error. Minimizing this expectation is equivalent to maximizing AUC. It is important to note that this justification is only valid under some strong assumptions, .i.e., (I) mismatch error $Z(\mathbf{O}) = Z$ is constant over all samples and (II) the multiclass AUC has similar properties to the binary AUC. Both of these assumptions may be violated in reality. Nevertheless, as we see in the experiments, the AUC maximization results in a robust weight estimation strategy.

To adapt AUC to our problem, we have to address three issues:

1. Generalize it to multiclass classification.
2. Transform HMM outputs to comparable scores.
3. Reduce the computational complexity of the AUC computation.

These issues are discussed in the following sections.

7.3.2 AUC Generalization to Multiclass Problem

For more than two classes, AUC is too complex to be calculated. With only 3 classes, the ROC curve should be plotted in a 6 dimensional space [Lan00]

and in general, the dimension of an ROC curve grows with square of the numbers of classes, $O(n^2 - n)$. To handle the multiclass situation, different methods have been suggested to approximate multiclass AUC including one-versus-all classes [PD00] method, pairwise AUC averaging [HT01] etc. Here, we used the one versus all approach due to its computational efficiency. For a k-class problem, it only needs to average over k AUC values:

$$A_{total} = \sum_{i=1}^{k} \pi_i A_i \tag{7.16}$$

where A_i is the AUC value of a binary classification task in which, the i^{th} class is considered the positive class and the rest to be negative. Equation (7.16) is an average of AUCs weighted by the class priors π_i. The disadvantage of this approach is that the AUC is now sensitive to class distributions.

7.3.3 Transforming HMM Outputs

Throughout the rest of this work, we assume M_1 to M_K are K given HMMs and the task is to assign a class label $k \in 1, ..., K$ to each sample (utterance) of the dataset. Furthermore, we assume the dataset is balanced. That is, the class frequencies are equal.

Rewritten from section 2, the log-likelihood of each sample can be seen as a linear function of reliability weights:

$$\log \hat{P}(\mathbf{O}|M_k) = \log P(\mathbf{O}, \mathcal{S}_*|M_k) = \mathbf{w}^\top \mathbf{p}_k \tag{7.17}$$

where the subscript k in $\mathbf{p}_k$ is to emphasize that $\mathbf{p}_k$ is the sum of the observation log-probabilities of model M_k. In the following, the $\hat{}$ of $\log \hat{P}$, which indicates that it is the Viterbi approximation of the log-likelihood, is dropped for the ease of notation. Since $\log P(\mathbf{O}|M_k)$ is a linear function of $\mathbf{w}$, reliability weight vector $\mathbf{w}$ can be interpreted as a linear classifier or one layer perceptron. This interesting fact, allows one to efficiently estimate the weight vector with respect to different criteria and more importantly to generalize the weight set to K weight sets, that is, one weight set for each pattern. This generalization is further explored in the following sections.

The score in (7.17) cannot be utilized as it is for AUC calculation. From (7.5) it is clear that for AUC computation, scores should be comparable while, $\log P(\mathbf{O}|M_k)$ is dependent on the observation probability $P(\mathbf{O})$ and thus

not comparable from one utterance to another. In the previous section we have used the logarithm of the likelihood ratio for the binary case (7.11). In multiclass case, however, there are various ways to define the likelihood ratio. Here we use the following definition for the multiclass likelihood ratio (MLR):

$$\text{MLR}_k = \frac{P(\mathbf{O}|M_k)^K}{\prod_{i=1}^{K} P(\mathbf{O}|M_i)} \tag{7.18}$$

This quantity is a generalization of the likelihood ratio in a binary classification task. Since it is independent of the sample probability $P(\mathbf{O})$, it is comparable among different samples and thus suitable for our framework. To make the classification decision, we need to compute K likelihood ratios for each sample. The final decision rule is that the sample belongs to the class with the maximum likelihood ratio value[4].

Now, by taking logarithm of MLR and replacing $\log P(\mathbf{O}|M_k)$ terms by their approximations $\mathbf{w}^\top \mathbf{p}_k$, we get:

$$\begin{aligned} L_k(\mathbf{O}) = \log \text{MLR}_k &= \mathbf{w}^\top (K\mathbf{p}_k - \sum_{i=1}^{K} \mathbf{p}_i) \\ &= \mathbf{w}^\top \mathbf{x}_{\mathbf{O}}^k \end{aligned} \tag{7.19}$$

where $\mathbf{x}_{\mathbf{O}}^k = K\mathbf{p}_k - \sum_{i=1}^{K} \mathbf{p}_i$ is simply the distance of the k^{th} log-likelihood of $\mathbf{O}$ from the average log-likelihoods of the K HMMs. The larger it is, the more likely it is that $\mathbf{O}$ belongs to class k. The obtained score $L_k(\mathbf{O})$ is independent of the sample probability meaning that given two samples $\mathbf{O}_1$ and $\mathbf{O}_2$ one may compare their scores $L_k(\mathbf{O}_1)$ and $L_k(\mathbf{O}_2)$ to conclude which one is represented better by model M_k. As we will see in the next section, this score will be used to compute the AUC.

7.3.4 Online AUC Maximization

The AUC metric is written as a sum of pairwise losses between samples from different classes. That is, the computational complexity of AUC is

[4]This also can be seen as a one-versus-all approach to convert a multiclass problem to K binary classification tasks. Other methods could be taken including pairwise comparisons or min-margin= $\frac{P(\mathbf{O}|M_k)}{\max_i P(\mathbf{O}|M_i)}$

quadratic in the number of samples. Consider a dataset $\mathcal{D} = \{(\mathbf{O}_i, y_i) \in \mathbb{R}^N \times \{1, \ldots, K\} | i \in BK\}$ with B samples from each of the K classes and $N = |A| + |V| + 1$. We split $\mathcal{D}$ into K subsets, $\mathcal{D}_k = \{(\mathbf{O}_i, y_i) | i \in \mathcal{D} \ \& \ y_i = k\}$. Combining (7.10) and (7.16), the AUC value for a multiclass classifier can be computed as:

$$\begin{aligned} A &= \frac{1}{Z} \sum_{k=1}^{K} \sum_{j \in \mathcal{D}_k} \sum_{n \notin \mathcal{D}_k} \mathbb{I}_{(L_k(\mathbf{O}_j) \geq L_k(\mathbf{O}_n))} \\ &= \frac{1}{Z} \sum_{k=1}^{K} \sum_{j \in \mathcal{D}_k} \sum_{n \notin \mathcal{D}_k} \mathbb{I}_{(\mathbf{w}^\top \mathbf{x}^k_{\mathbf{O}_j} \geq \mathbf{w}^\top \mathbf{x}^k_{\mathbf{O}_n})} \\ &= \frac{1}{Z} \sum_{k=1}^{K} \sum_{j \in \mathcal{D}_k} \sum_{n \notin \mathcal{D}_k} 1 - \mathbb{I}_{(\mathbf{w}^\top \mathbf{x}^k_{\mathbf{O}_j} \leq \mathbf{w}^\top \mathbf{x}^k_{\mathbf{O}_n})} \end{aligned} \tag{7.20}$$

where Z is a normalization factor. Maximizing AUC is equivalent to minimizing the summation of $\mathbb{I}_{(\mathbf{w}^\top \mathbf{x}^k_{\mathbf{O}_j} \leq \mathbf{w}^\top \mathbf{x}^k_{\mathbf{O}_n})}$ terms in (7.20). Unfortunately, this is a combinatorial optimization problem due to the non-convexity of the indicator function. In order to obtain a convex optimization problem, a common approach is to approximate the indicator function with its convex surrogate. Following the suggestion in [ZJYH11], we may replace the indicator function with the hinge loss function,

$$\ell(\mathbf{w}, \mathbf{x}^k_{\mathbf{O}_j} - \mathbf{x}^k_{\mathbf{O}_n}) = \max\left(0, 1 - \mathbf{w}^\top (\mathbf{x}^k_{\mathbf{O}_j} - \mathbf{x}^k_{\mathbf{O}_n})\right) \tag{7.21}$$

as shown in Figure 7.2. However, a drawback of this convex function, and in general any other convex loss, is that it assigns large loss values to outliers making the algorithm to only focus on outliers. This phenomenon has been detected and extensively studied in the context of boosting classifiers [LS10] which in some sense is a dual form of online learning [FS97] that we use here. As a remedy, we suggest to use a hinge loss with level threshold,

$$\begin{aligned} &\ell(\mathbf{w}, \mathbf{x}^k_{\mathbf{O}_j} - \mathbf{x}^k_{\mathbf{O}_n}) \\ &\quad = \min\left(T_0, \max\left(0, 1 - \mathbf{w}^\top (\mathbf{x}^k_{\mathbf{O}_j} - \mathbf{x}^k_{\mathbf{O}_n})\right)\right) \end{aligned} \tag{7.22}$$

where T_0 is the cut-off threshold as depicted in Figure 7.2. The loss function is now non-convex and the employed online minimization algorithm cannot guarantee to achieve the global optimum.

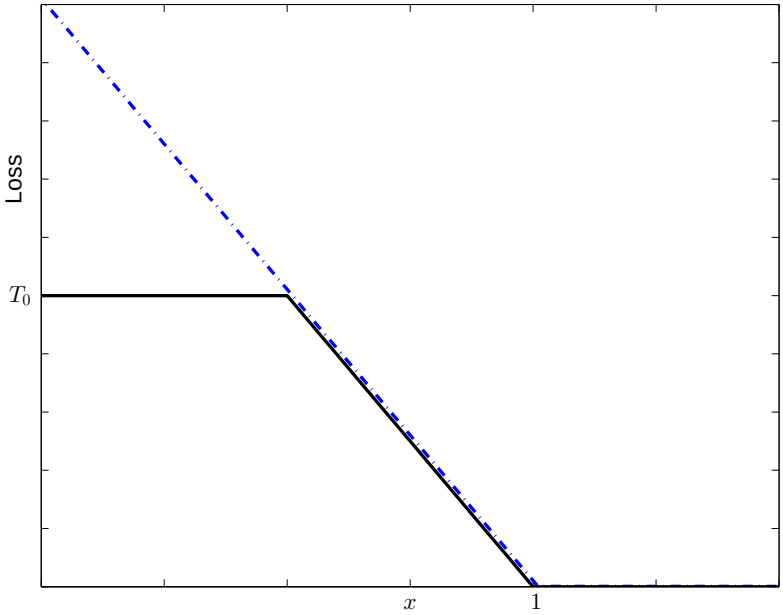

Figure 7.2: Illustrating hinge function and its non-convex thresholded version.

Having (7.20) and (7.22) in hand, the AUC maximization problem can be formulated as:

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||_2^2 + C \sum_{k=1}^{K} \sum_{j \in \mathcal{D}_k} \sum_{n \notin \mathcal{D}_k} \ell(\mathbf{w}, \mathbf{x}^k_{\mathbf{O}_j} - \mathbf{x}^k_{\mathbf{O}_n}) \tag{7.23}$$

$$\sum_i w_i = 1 \quad , \quad 0 \leq w_i \quad \forall i \in \{0, \ldots, |A| + |V|\}$$

where $\frac{1}{2}||\mathbf{w}||_2^2$ is a quadratic regularization term added to control the solution complexity. Other regularization terms such as entropy function or L1 norm regularization are also commonly used in the literature. Moreover, C is a penalty parameter of the error term. The constraints in (7.23) guarantee that the solution lies on the probability simplex.

The constraint set, which may be unnecessary for information fusion, is useful for modeling the inter-feature correlations. On a probability simplex,

if one weight increases the sum of the rest of the weights decreases. So projecting onto a probability simplex tries to alleviate the violation of the independence assumption by correlating the reliability weights to each other. It is conceivable thus that projecting to different convex sets results in different level of correlation modeling. For instance, projecting onto a unit hypercube ($0 \leq w_i \leq 1$) gives less credit to the features' correlations and projecting onto $\mathbb{R}^+$ ($0 \leq w_i$) assumes almost no correlation among features.

Irrespective of the constraints, the minimization in (7.23) has two important properties. It is a batch minimization and it is quadratic in the number of samples. These properties raise two issues. First, as the size of the training data increases, the computational cost quadratically increases making it prohibitive for large datasets. Second, by having new training material, the minimization problem should be resolved and weights cannot be updated. In other words, this framework is not suitable for adaptive weight adjustment.

We require an online optimization framework with low computational and memory complexity that can be updated as the new samples become available. To this end, we utilize the online AUC maximization framework, suggested in [ZJYH11] with some small modifications to fit our multiclass problem.

Since the standard online optimization framework is the sum of the losses of individual samples, we rewrite (7.23) to fit this framework, i.e.,

$$\frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{t=1}^{T}\mathcal{L}_t(\mathbf{w}) \tag{7.24}$$

where,

$$\begin{aligned}\mathcal{L}_t(\mathbf{w}) &= \mathbb{I}_{(y_t=1)}h_1^t(\mathbf{w}) + \cdots + \mathbb{I}_{(y_t=K)}h_K^t(\mathbf{w}) \\ h_i^t(\mathbf{w}) &= \sum_{t'=1}^{t'=t-1}\mathbb{I}_{(y_{t'}\neq i)}\ell(\mathbf{w}, \mathbf{x}^i_{\mathbf{O}_t} - \mathbf{x}^i_{\mathbf{O}_{t'}}) \quad i \in \{1, ..., K\}\end{aligned} \tag{7.25}$$

As T tends towards infinity, equation (7.24) approaches the objective function in (7.23) since it is an unbiased estimate of it. The projected gradient decent algorithm introduced in [Zin03] can at this step be utilized to solve this optimization problem. The weight update strategy based on the projected gradient descent algorithm is listed in Algorithm 7.2.

In Algorithm 7.2, proj(.) is a projection function, $\triangle$ is the probability

Algorithm 7.2: Single Vector: Projected Gradient Descent

Input: set $\mathbf{w}^1 = [\frac{1}{N}, \ldots, \frac{1}{N}]^\top$.
for $t = 1, \ldots, T$ **do**
receive $(\mathbf{O}_t, y_t)$ $\mathbf{w}^{t+1} = \text{proj}_\triangle\Big(\mathbf{w}^t - \eta \frac{\partial}{\partial \mathbf{w}} \mathcal{L}_t(\mathbf{w})\Big)$
end
Output: The final weight vector $\mathbf{w}^T$.

simplex and η is a learning rate. Projection onto probability simplex is defined as $\text{proj}_\triangle(\mathbf{z}) = \underset{\mathbf{x} \in \triangle}{\text{argmin}} ||\mathbf{z} - \mathbf{x}||_2^2$.

As can be seen, the projected gradient decent algorithm needs to compute the (sub)gradient of $\mathcal{L}_t(\mathbf{w})$ at each update round[5]. For thresholded hinge loss function for instance, the subgradient with respect to $\mathbf{w}$ when $y_t = i$ is:

$$\frac{\partial}{\partial \mathbf{w}} \mathcal{L}_t(\mathbf{w}) = \sum_{t'=1}^{t'=t-1} (\mathbf{x}^i_{\mathbf{O}_{t'}} - \mathbf{x}^i_{\mathbf{O}_t}), \tag{7.26}$$

when $-1 \leq \mathbf{w}^\top (\mathbf{x}^i_{\mathbf{O}_{t'}} - \mathbf{x}^i_{\mathbf{O}_t}) \leq T_0$, and zero otherwise. Since to compute this subgradient all the previous samples are required, Zhao. et al. [ZJYH11] introduced a buffer based algorithm to only store the last few samples to be used for subgradient computation. Furthermore, they showed that the difference between the optimal solution and the solution obtained by the buffer based method is bounded. Although in this work, we set the buffer size to be the same size as the number of samples in the training data, using this trick can be quite practical for large datasets or when it is required to update the weights in real time.

7.4 Multi-vector Model

As mentioned before, interpreting the weight vector as a linear classifier enables us to further generalize this notion. Here we use the multi-vector generalization of linear classifier to address the multiclass problem. In this setting, an individual weight vector is dedicated to each class, i.e., K different $\mathbf{w}$ in our problem. Let us denote the weight matrix $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K]$,

[5] In our case it is in fact subgradient since hinge loss is not differentiable.

where $\mathbf{w}_i$ is the weight vector of HMM M_i. While the main body of the proposed algorithm for the single-vector case remains intact, there are still some modifications required. As discussed in section 7.3, when AUC is used as a criterion, the scores of utterances should be comparable, that is, they should be independent of the observation probability $P(\mathbf{O})$. In the single-vector case, it is sufficient to use MLR in (7.18) to satisfy this requirement. However, in the multi-vector setting where each $P(\mathbf{O}|M_k) = \mathbf{w}_k\mathbf{p}_k$ has its own weight vector, $P(\mathbf{O})$ will not be removed from the equation (7.18) (since it is weighted with different values) and thus, MLR is not $P(\mathbf{O})$-independent anymore. Thus, in the multi-vector case, instead of MLR we use a heuristic to achieve approximately $P(\mathbf{O})$-independent scores. Given an observation sequence $\mathbf{O}_r = \{\mathbf{o}^1, ..., \mathbf{o}^{T_r}\}$ with length T_r, we normalize its corresponding likelihood vector $\mathbf{p}$ by the number of observations T_r to make the score of a given sequence independent of its length. Similar to the single-vector case in (7.19), the score is,

$$L_k(\mathbf{O}_r) = \frac{1}{T_r}\mathbf{w}_k^\top(K\mathbf{p}_k - \sum_{i=1}^{K}\mathbf{p}_i) = \mathbf{w}_k\mathbf{x}_{\mathbf{O}_r}^k \tag{7.27}$$

where $\mathbf{x}_{\mathbf{O}_r}^k = \frac{1}{T_r}(K\mathbf{p}_k - \sum_{i=1}^{K}\mathbf{p}_i)$. Having this modification in hand, it is straightforward to generalize the Algorithm 2 to the multi-vector case. To this end, let us rewrite (7.27) in the matrix-trace form:

$$L_k(\mathbf{O}_r) = \mathrm{Tr}(\mathbf{D}_k\mathbf{W}^\top\mathbf{X}_{\mathbf{O}_r}) \tag{7.28}$$

where $\mathrm{Tr}(.)$ stands for the trace of a matrix, $\mathbf{D}_k$ is a diagonal matrix with only one non-zero element in its k^{th} diagonal position set to be 1 and $\mathbf{X}_{\mathbf{O}_r} = [\mathbf{x}_{\mathbf{O}_r}^1, ..., \mathbf{x}_{\mathbf{O}_r}^K]$. Then, the AUC optimization (7.23) in multi-vector case is:

$$\min_{\mathbf{W}} \frac{1}{2}||\mathbf{W}||_2^2 + C\sum_{k=1}^{K}\sum_{j\in\mathcal{D}_k}\sum_{n\notin\mathcal{D}_k}\ell(\mathbf{W}, \mathbf{x}_{\mathbf{O}_j}^k - \mathbf{x}_{\mathbf{O}_n}^k) \tag{7.29}$$

$$\text{s.t} \qquad \mathbf{W}^\top\mathbf{1}_{(|A|+|V|+1)\times 1} = \mathbf{1}_{K\times 1} \quad , \quad \{W_{i,j}\} \geq 0$$

where $\mathbf{1}$ is an all-one vector, $\{W_{i,j}\}$ stands for all entries of $\mathbf{W}$, $||\mathbf{W}||_2^2$ is defined as $\mathrm{Tr}(\mathbf{W}^\top\mathbf{W})$ and,

$$\begin{aligned}&\ell(\mathbf{W}, \mathbf{x}_{\mathbf{O}_j}^k - \mathbf{x}_{\mathbf{O}_n}^k)\\ &= \min\Big(T_0, \max\big(0, 1-\mathrm{Tr}(\mathbf{D}_k\mathbf{W}^\top(\mathbf{X}_{\mathbf{O}_j} - \mathbf{X}_{\mathbf{O}_n}))\big)\Big)\end{aligned} \tag{7.30}$$

Similar to the single-vector case, this optimization can be solved by using the projected gradient decent algorithm. For the t^{th} sample belonging to i^{th} class, the subgradient is equal to:

$$\begin{aligned}\frac{\partial}{\partial \mathbf{W}}\mathcal{L}_t(\mathbf{W}) &= \sum_{t'=1}^{t'=t-1} \frac{\partial}{\partial \mathbf{w}} \mathrm{Tr}\big(\mathbf{D}_k \mathbf{W}^\top (\mathbf{X}_{\mathbf{O}'_t} - \mathbf{X}_{\mathbf{O}_t})\big) \\ &= \sum_{t'=1}^{t'=t-1} [\mathbf{0}, ..., \mathbf{0}, \mathbf{x}^i_{\mathbf{O}_{t'}} - \mathbf{x}^i_{\mathbf{O}_t}, \mathbf{0}, ..., \mathbf{0}], \end{aligned} \tag{7.31}$$

when $-1 \leq \mathrm{Tr}\big(\mathbf{D}_k \mathbf{W}^\top (\mathbf{X}_{\mathbf{O}'_t} - \mathbf{X}_{\mathbf{O}_t})\big) \leq T_0$, and zero otherwise. In (7.31), columns $\mathbf{0}$ denote all-zero vectors. As seen, all columns of the subgradient matrix except one, the i^{th} column, are zero. Thus at each update step t only one column of $\mathbf{W}$ is updated.

Algorithm 7.3: Multi Vector: Projected Gradient Descent

Input: set $\mathbf{W}^1 = [\frac{1}{N}\mathbf{1}, \ldots, \frac{1}{N}\mathbf{1}]$.
for $t = 1, \ldots, T$ **do**
receive $(\mathbf{O}_t, y_t)$, set $i = y_t$
$\mathbf{W}_i^{t+1} = \mathrm{proj}_{\triangle}\Big(\mathbf{W}_i^t - \eta \frac{\partial}{\partial \mathbf{W}} \mathcal{L}_t(\mathbf{W})|_{i^{\text{th}}\text{column}}\Big)$
end
Output: The final weight matrix $\mathbf{W}^T$.

In algorithm 7.3, $\mathbf{W}_i^t$ denotes the i^{th} column of $\mathbf{W}$ at round t and subgradients are coimputed as in (7.31). Comparing Algorithm 2 and 3 reveals that both single- and multi-vector cases have equal computational complexity. However, it is important to note that in the multi-vector case, $\mathbf{W}_i$ is on average T/K times updated after T iterations and thus, its convergence is K times slower than the single-vector case.

7.5 Experiments on AUC-based Fusion Strategy

In the following experiments, the proposed weighting estimation scheme is evaluated in different scenarios. The sound units are words and the task at hand is to develop a robust audio-visual digit recognition system. Here, the CUAVE dataset [PGTG02] was used, which contains the digits from zero to

nine repeated five times by 36 speakers (see Section 2.2.1). In order to have a speaker-independent evaluation, the one-speaker-out cross-validation strategy was utilized. The audio signals were corrupted by additive white noise at various SNR levels $\{-20, -15, -10, -5, 0, 5, 10, 15, \infty\}$ dB, while video signals were clean in all experiments.

For digit recognition, we used a similar GMM-HMM based V-ASR introduced in Chapter 6. As in that Chapter, the ISMA features which consists of COBRA-selected ISCN features and 10 posterior probabilities of MuMABoost were considered to be visual features. Compared with Chapter 3, here a slightly smaller subset of ISCN features, i.e., 30 features, were selected. By including the first- and second-order derivatives of the visual features, the final 120 visual features used to train GMM-HMMs were obtained.

As audio features, the mel-frequency cepstral coefficients (MFCC) were used. Thirteen MFCC features and their first- and second-order derivatives were extracted from frames with 20 ms duration and 10 ms overlap. That is, the audio frame-rate was 100 frames per second. Visual features were linearly interpolated to increase their frame-rate from 60 to 100 to have equal number of audio and visual features per utterance. Finally, both audio and visual features were normalized per speaker by subtracting their mean values for each speaker.

Audio-visual feature vectors were employed to train 10 HMMs; one HMM per digit. Each Markov state in HMMs was modeled with two Gaussian components with diagonal covariance matrices. The number of emitting states in all HMMs was empirically chosen to be nine. Moreover, the popular HTK toolkit [YEG+06] was used to train the HMMs. It is important to remark that during training audio and visual features had equal weights.

In the first experiment, the accuracy of digit classification was evaluated by using three weighting strategies: (I) Optimal strategy, (II) Minimizing the classification error and (III) Proposed strategy.

The optimal strategy needs a priori knowledge about test conditions. In this setting, the weights are selected in order to minimize the classification error rate on each particular test data, i.e., an individual weight set for each SNR value.

The second strategy is to estimate the weights by minimizing the error rate over the *clean* training data. This strategy is called min-error in Figure 7.3 and its performance over different SNR values is depicted with a solid red line.

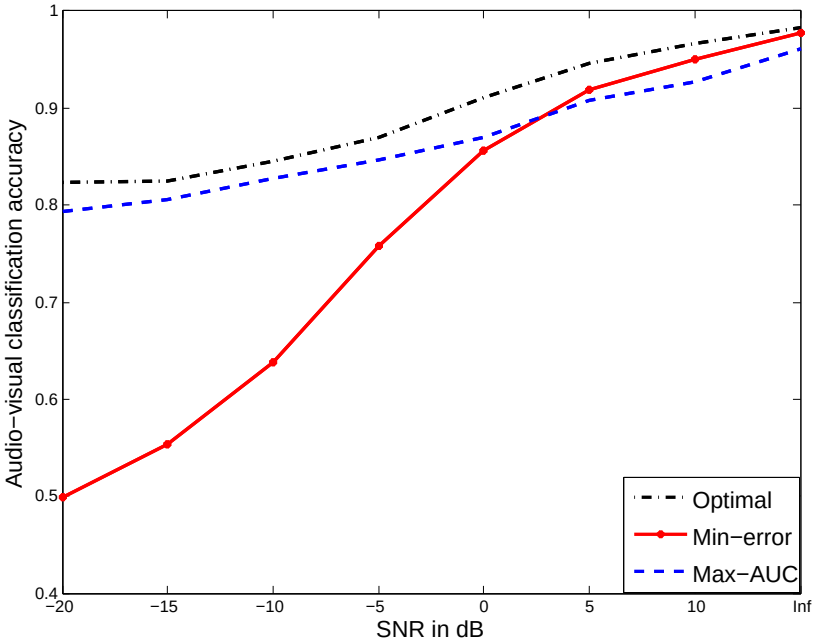

Figure 7.3: Comparison between optimal (no mismatch), maximum AUC and minimum classification error based strategies for estimating the feature weights.

The proposed strategy (in single-vector case), which is called max-AUC in Figure 7.3, chooses the reliability weights such that the approximation of AUC over the *clean* training data is maximized. The first two methods serve as reference to compare with our approach.

As seen in Figure 7.3, the proposed strategy is very robust against the mismatch conditions and works uniformly well over different SNR regions while min-error approach shows a significant performance degradation as the SNR value decreases. This figure clearly shows that employing an appropriate criterion to estimate the weights is essential to achieve robustness against varying conditions. Figure 7.3 summarizes the main contribution of this work. That is, employing a set of fixed reliability weights that have been chosen in accordance with a robust criterion, can achieve close-to-optimal accuracy in a wide range of SNR values. In the second set of experiments reported in

Modalities	Audio-only	Video-only	Audio-visual
		∞ dB	
159W	98.00	81.69	96.35
2W	97.72	81.35	95.38
159W-match	98.00	81.69	96.35
2W-match	97.72	81.35	95.38
		10 dB	
159W	93.08	81.69	93.04
2W	92.81	81.35	91.64
159W-match	93.53	81.69	94.11
2W-match	92.81	81.35	91.92
		5 dB	
159W	83.94	81.69	91.14
2W	84.28	81.35	89.61
159W-match	84.26	81.69	91.54
2W-match	84.28	81.35	90.00
		0 dB	
159W	65.17	81.69	87.74
2W	64.26	81.35	86.51
159W-match	67.15	81.69	89.14
2W-match	64.26	81.35	86.51

Table 7.1: Classification rates in different noise levels for single-vector setting. 159W rows report the accuracies for one weight per feature (159) case and 2W rows are for one weight per modality. 159W-match and 2W-match represent the scenario where the reliability weights are estimated from match-to-test datasets.

Tables 7.1 and 7.2, the ten HMMs were trained with *clean* audio-visual data and tested in different noisy conditions. The weights were estimated by maximizing AUC with the non-convex loss for two cases: one weight for each feature (159 in total) which we refer to as 159W in our tables and one weight for each stream (only 2 weights) called as 2W. Moreover, we also report the results of a scenario where the weights were estimated from match-to-test data, called "159W-match" and "2W-match" in Tables 7.1 and 7.2 . The match-to-test data was constructed by randomly selecting 10% of the training data and adding appropriate additive noise to it to match the test scenario.

Modalities	Audio-only	Video-only	Audio-visual
		-5 dB	
159W	39.68	81.69	85.26
2W	36.85	81.35	84.18
159W-match	44.08	81.69	85.53
2W-match	36.85	81.35	84.18
		-10 dB	
159W	20.22	81.69	82.24
2W	17.46	81.35	82.00
159W-match	24.21	81.69	83.57
2W-match	17.46	81.35	82.06
		-15 dB	
159W	12.47	81.69	80.58
2W	11.08	81.35	80.67
159W-match	14.32	81.69	82.14
2W-match	11.08	81.35	81
		-20 dB	
159W	8.889	81.69	79.05
2W	8.333	81.35	79.78
159W-match	8.667	81.69	81.56
2W-match	8.333	81.35	80.06

Table 7.2: Classification rates in different noise levels for single-vector setting. 159W rows report the accuracies for one weight per feature (159) case and 2W rows are for one weight per modality. 159W-match and 2W-match represent the scenario where the reliability weights are estimated from match-to-test datasets.

The motivation for running this experiment was to estimate, given a hypothetical adaptive weight estimation algorithm (that can adaptively update the reliability-weights to match test conditions), how much additional classification accuracy may be obtained? In reality, however, an adaptive algorithm can never precisely estimate the test conditions. Thereby, the improvements reported in Tables 7.1 and 7.2 for the match case should be interpreted as being overly optimistic. The classification accuracies for audio-only, video-only and audio-visual classification systems are reported in the second to forth columns of Tables 7.1 and 7.2, respectively.

Modalities	Audio-only	Video-only	Audio-visual
		∞ dB	
159W	96.96	82.92	96.69
2W	97.06	78.44	94.57
159W-match	96.96	82.92	96.69
2W-match	97.06	78.44	94.57
		10 dB	
159W	91.49	82.92	96.53
2W	92.06	78.44	93.00
159W-match	92.28	82.92	96.58
2W-match	92.06	78.44	93
		5 dB	
159W	84.97	82.92	95.29
2W	85.58	78.44	91.21
159W-match	88.03	82.92	95.61
2W-match	85.58	78.44	91.26
		0 dB	
159W	73.04	82.92	93.4
2W	74.01	78.44	88.78
159W-match	79.37	82.92	93.85
2W-match	73.96	78.44	88.78

Table 7.3: Classification rates (in percentage) at different noise levels for multi-vector case. 159W rows report the accuracies for one weight per feature (159) case and 2W rows are for one weight per stream case. 159W-match and 2W-match represent the scenario where the reliability weights are estimated from match-to-test datasets.

Two interesting trends can be detected in Tables 7.1 and 7.2. First, the one weight per feature case leads to 1% to 2% classification rate improvement in high-SNRs regimes, i.e., above -5 dB. However, as the SNR value declines, the advantage of using a large number of weights (here 159) decreases and at -15 dB it is observed that one weight per modality results in a slightly better performance than the 159W case. That is, the mismatch between training and test conditions may more severely affect a model with 159 hyper-parameters than a model with only 2 hyper-parameters.

The second trend detected in Tables 7.1 and 7.2 is that in the match case

Modalities	Audio-only	Video-only	Audio-visual
		-5 dB	
159W	52.18	82.92	89.14
2W	54.46	78.44	84.69
159W-match	62.68	82.92	90.08
2W-match	54.35	78.44	84.75
		-10 dB	
159W	31.10	82.92	82.99
2W	34.29	78.44	80.13
159W-match	44.85	82.92	86.42
2W-match	34.53	78.44	80.18
		-15 dB	
159W	18.65	82.92	77.10
2W	21.67	78.44	75.47
159W-match	29.40	82.92	81.71
2W-match	21.72	78.44	75.36
		-20 dB	
159W	11.94	82.92	70.53
2W	13.58	78.44	72.07
159W-match	20.31	82.92	78.10
2W-match	13.58	78.44	72.18

Table 7.4: Classification rates (in percentage) at different noise levels. 159W rows report the accuracies for one weight per feature (159) case and 2W rows are for one weight per stream case. 159W-match and 2W-match represent the scenario where the reliability weights are estimated from match-to-test datasets.

scenario, the one weight per stream (2W case) does not show a meaningful improvement over the 2W case which may seem surprising. In fact, in most cases, accuracies obtained by 2W-match are very close to 2W cases. The ineffectiveness of the 2W setting to adapt to a particular condition is due to the fact that it does not introduce sufficient degrees of freedom into the model to properly present the complex reality. However, in the 159W-match cases, where the weights were estimated from a match-to-test training set, significant improvements in all SNRs over the mismatch cases can be observed. That is, given an adaptive algorithm that can update the weights in order to

match test conditions, it is more rewarding to employ a larger set of weights than only two stream weights.

In the third experiment reported in Tables 7.3 and 7.4, we evaluated the performance of the multi-vector setting where 10 different weight vectors are estimated; one for each digit. As in the second experiment, 159W represents the one weight per feature case and 2W stands for the one weight per modality. An immediate observation is that the multi-vector setting greatly outperforms the single-vector setting in a wide range of SNRs from 10 dB to -10 dB. It seems this improvement is due to the fact that the multi-vector setting significantly improves the quality of the audio-only recognizer. For instance, at -5 dB the audio-only performance for 159W in the single-vector setting is 39% while for multi-vector setting it reaches to 52%.

As in the single-vector setting, it is observed that 159W is more sensitive to mismatch error and achieves 2% less accuracy rate at -20 dB than the 2W case. However, it consistently shows better performance in higher SNRs.

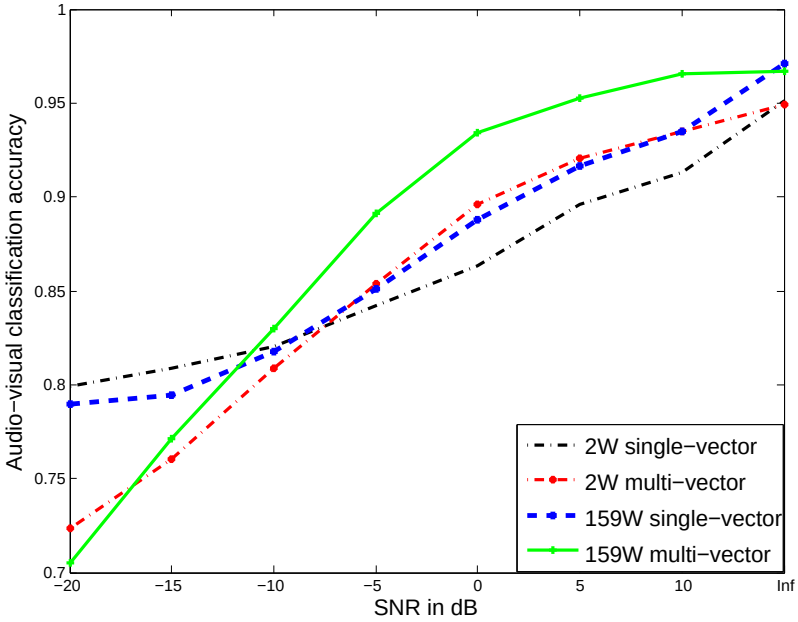

Figure 7.4: Comparison between the 4 proposed weight sets.

Figure 7.4 compares all the four cases, i.e., 159W single-vector, 2W single-vector, 159W multi-vector and 2W multi-vector. From this figure, it is clearer which setting is more suitable for which SNR region(s). The underlying principle is that as the number of parameters increases, both the accuracy of a model and the sensitivity to mismatch error increase. The different methods in Figure 7.4 offer different compromises between these two factors.

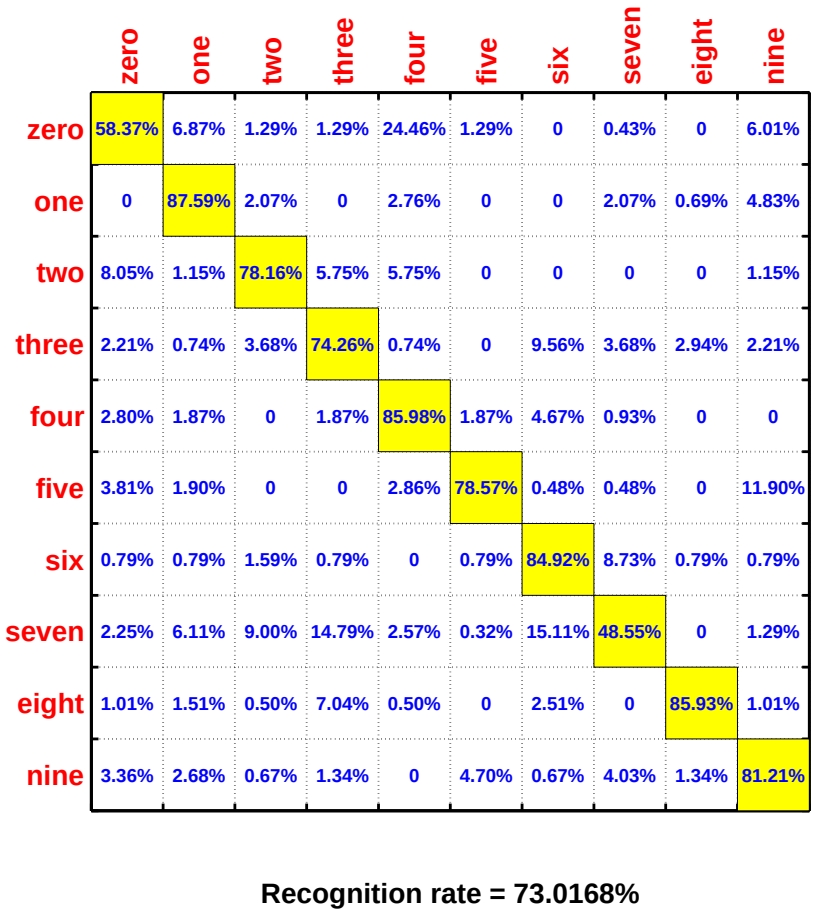

Figure 7.5: Confusion matrices of audio-based classifier for multi-vector 159W setting in 0 dB.

An important question raised here is whether the audio and visual-based classifiers often make mistakes on similar test samples or their errors are roughly uncorrelated. In other words, are the *hard-to-classify* samples for both classifiers similar?

Figures 7.5, 7.6 and 7.7 depict the confusion matrices of audio, visual and audio-visual based classifiers for multi-vector 159W setting in 0 dB, respectively.

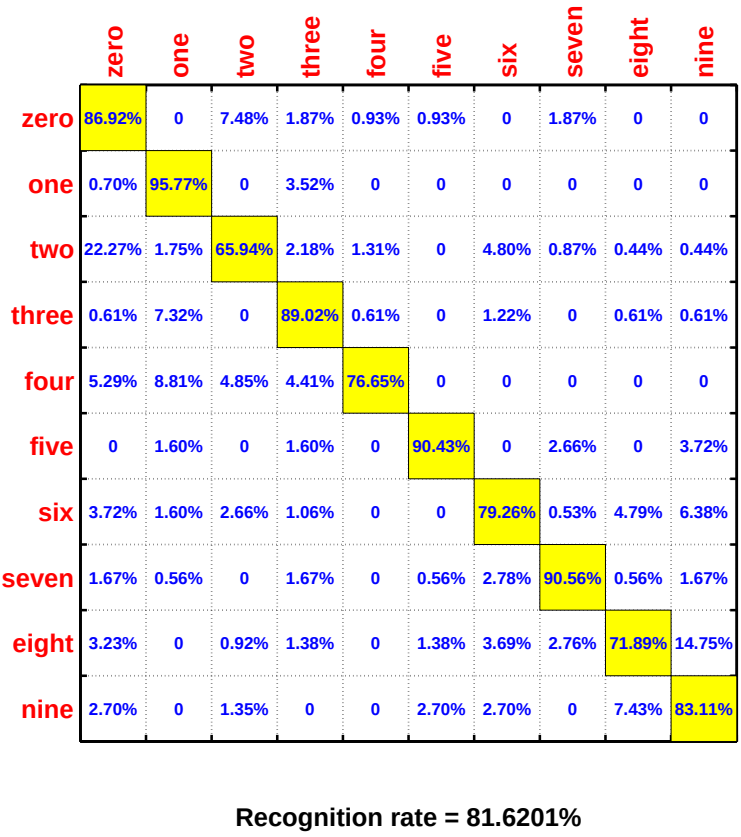

Figure 7.6: Confusion matrices of visual-based classifier for multi-vector 159W setting in 0 dB.

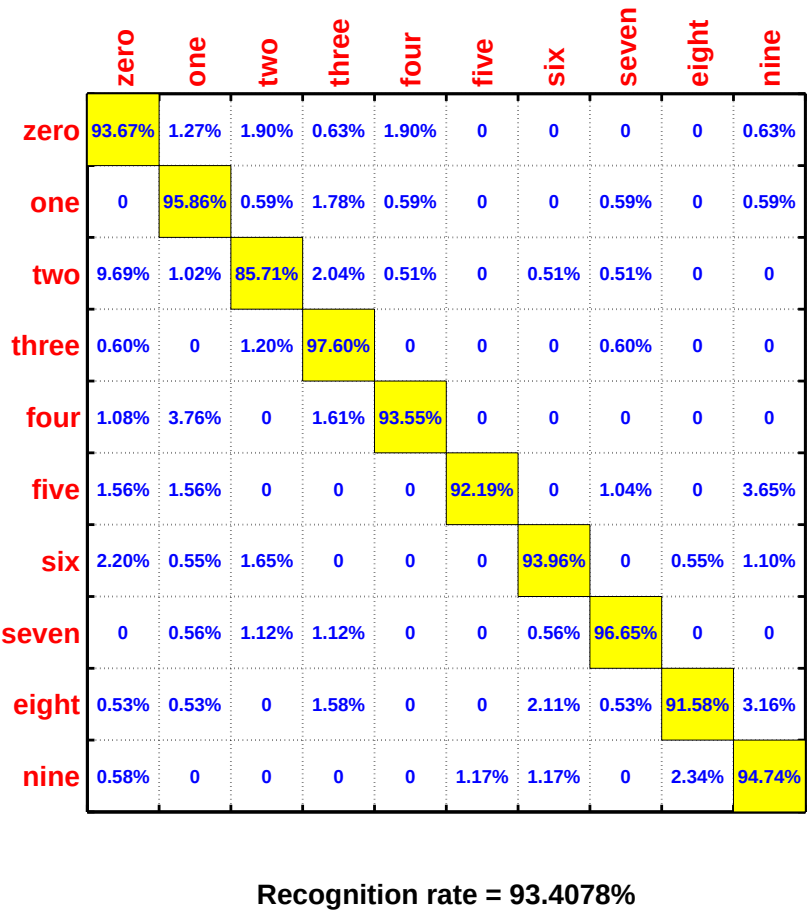

Figure 7.7: Confusion matrices of audio-visual classifier for multi-vector 159W setting in 0 dB.

As seen, digit *four* was the most difficult class to classify for audio-based recognizer and it only classified 51.4% of the samples belonging to this class correctly. On the other hand, samples from *four* were the easiest to classify for visual-based classifier. It yielded 97% classification accuracy on that class. Another interesting result is for digit *nine*. Both classifiers give

almost the same accuracy for this class, i.e., 67-68%. However, while audio classier considers most of the wrongly classified samples of this class to be *five*, visual-based classifier assigns most of the wrongly classified samples to class *seven*. The accuracy improvement of the combined classifier over the single-modality classifiers depends on the level of the correlation between the classifiers (see random forest analysis in [Bre01]). The weaker the correlation between the outputs of the classifiers, the higher the accuracy of the combined classifier is. This explains the classification improvement of the audio-visual classifier over the audio-only or visual-only classifiers.

7.6 Conclusion

In order to achieve robustness against training and testing mismatch, we proposed to employ AUC as a design criterion to estimate the reliability weights. We showed that the reliability weight vector could be interpreted as a linear classifier taking the likelihood values computed by HMMs as an input feature vector. We evaluated different weight sets such as one weight per stream, one weight per feature and one weight vector per class. As shown in the experiments, except for the severe mismatch condition (-20 dB), the one weight per feature scenario outperforms the one weight modality both in multi-vector and single-vector setting. Multi-vector setting, which dedicates an individual weight vector to each class, showed a considerable improvement over the single-vector case in a wide range of SNR values. It however, seems to be more sensitive to mismatch error as its accuracy rate in -20 dB is about 10% less than the single-vector case. With an adaptive model selection algorithm, a multi-mode fusion algorithm may employ all these four models to achieve an optimal accuracy over all SNR regions.

Chapter 8

Multichannel Audio-video Speech Recognition System

This section is devoted to multichannel audio-visual speech recognition systems. The AV-ASR system proposed in this chapter is almost similar to the system introduced in Chapter 6 with one additional audio-processing block, beamforming. Since audio signals are captured by a microphone array with eight channels, we can utilize beamforming techniques to focus on the desired sound source and alleviate background noise. The output of the beamforming block is then supplied to the AV-ASR discussed in Chapters 6 and 7.

The multichannel audio-visual dataset used to evaluate the proposed algorithms in this Chapter is collected in a highly reverberant office room, under highly varying light conditions and with a relatively low-resolution RGB camera. Using this dataset to evaluate the proposed algorithms, yields a fair estimate of the performance of the algorithms in adverse conditions which commonly occur in real-world applications.

8.1 Introduction to Beamforming Problem

While visual information is of a high importance for increasing the robustness of AV-ASR in several respects, it hardly can help (at least directly) to improve the speech signals in reverberant environments. Reverberation corrupts harmonic structure in voiced speech and consequently decreases the speech

recognition rate. One simple approach to overcome this difficulty is to exploit beamforming techniques, to focus on the desired direction and attenuate the echoes of the signal.

However, although this solution may work in fixed (non-adaptive) beamformers, it leads to signal cancellation in adaptive beamformers, due to a phenomenon known as signal leakage. Adaptive beamformers tend to adapt to the real characteristics of the background noise rather than a predetermined model like white or diffuse noise. To this end, they estimate the statistics of background noise from the non-speech segments of audio signals. However, in a reverberant room, the echoes of the speech signals are still sensed by microphones long they end. These echoes bear the same statistical signature as the desired signal and hence result in signal cancellation when they are considered as noise by beamformer.

The commonly utilized minimum variance distortion-less response (MVDR) beamformer minimizes the output power while maintaining a specified response to the desired signal. However, in the presence of coherent interferers, which occur due to reverberation, microphone mismatch or steering vector error, MVDR fails due to the signal leakage problem. Signal processing methods that have been proposed to address this problem are usually known as robust adaptive beamforming methods. However, this robustness is achieved at the expense of less interference reduction or an increased number of microphones. For example in [SK85], the signal is averaged over the space to decorrelate the signal and interference. This technique can only be applied to uniform microphone arrays [1] and needs many microphones to achieve satisfactory performance. Using norm-constrained adaptive filters was proposed in [QV95] to constrain the power of the signal leakage and which consequently leads to improving the adaptive beamformers. It, however, requires knowledge of the interference covariance matrix which may not be available in speech recognition applications. In [HSH99], both quadratic and non-linear (truncation) constraints in three-block structure have been used to improve the interference reduction. Another approach is to estimate the transfer functions (TFs) with blind source separation techniques. TFs may also be estimated based on non-stationarity of signal and stationarity of noise assumption [GBW01].

[1] A uniform microphone array is a linear microphone array with equal distances between the microphones.

In this work, we extract the TFs from the covariance matrix of the array data given the source locations. we show that this problem can be formulated as an instance of weighted Procrustes problem [Vik06]. Typically, a Procrustes problem is used to rotate and scale a set of data to fit another set. Here this method is used to estimate those TFs that are close enough to the ideal TFs (steering vectors) and still can reconstruct the data covariance matrix.

Using the proposed beamforming method yields 2% performance improvement in audio-only speech recognizer when microphone array consists of 8 microphones and obtains 1.11% when 4-channel microphone array is utilized.

8.2 Signal Cancellation

Let $s_m(n)$ be the sound waves emitted by M wide-band sources which are received by an array of N microphones. The room impulse response $h^{k,i}_{room}(t)$ characterizes both direct and echo paths from the k-th source to the i-th microphone. Since the microphones may not be calibrated and may introduce different transfer functions $h^i_{mic}(t)$, one can merge the room impulse response and microphone transfer function into a total transfer function $h^{k,i}(t)$ containing both room acoustic and microphone characteristics. Therefore the i-th microphone output in the frequency domain can be written as:

$$x_i(f) = \sum_{m=1}^{M} h^{k,i}(f)s_m(f) + n_i(f), \tag{8.1}$$

where $n_i(f)$ is the additive noise at the i-th microphone. The microphone outputs can be aggregated into a column vector $\mathbf{x}$:

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{n}, \tag{8.2}$$

where $\mathbf{n}$ is the additive white noise vector, $\mathbf{H} = [\mathbf{h}_1, ..\mathbf{h}_M]$ is the channel matrix, $\mathbf{h}_i = [h^{k,i}, ..., h^{M,i}]^T$ the i-th column of $\mathbf{H}$ and $\mathbf{s} = [s_1(f), ..., s_M(f)]$ is the source vector. The number of sources M is assumed to be less than the number of microphones N. Without loss of generality, s_1 can be considered as the desired signal. The goal of the beamformer is to obtain an estimate of the desired signal by filtering and summing the microphone outputs:

$$y(f) = \mathbf{w}(f)^H\mathbf{x}(f), \tag{8.3}$$

where $(.)^H$ is the Hermitian transpose operator, y the beamformer output and $\mathbf{w}$ the beamformer weight vector. The conventional MVDR beamformer chooses its weight vector $\mathbf{w}$ to minimize the output power while maintaining the signal from a specified direction of arrival:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \quad \mathbf{w}^H \mathbf{R} \mathbf{w} \qquad \text{subject to} \quad \mathbf{w}^H \mathbf{d} = 1 \tag{8.4}$$

$\mathbf{R} = E\{\mathbf{x}\mathbf{x}^H\}$ is the covariance matrix of received signals $\mathbf{x}$. The target steering vector is defined by $\mathbf{d} = [e^{-j\omega\tau_1}, ..., e^{-j\omega\tau_N}]$, where $\tau_1 ... \tau_N$ are delays matched to the desired speaker direction. The conventional MVDR presented in (8.4), however, can only perform well in anechoic environments where noise and signal are independent. In reality, severe signal cancellation occurs because of microphone mismatch, location estimate errors, signal-correlated noise and reverberant environments. To clarify this problem, we have to look more closely into the covariance matrix $\mathbf{R}$:

$$\mathbf{R} = E\{\mathbf{x}\mathbf{x}^H\} = \mathbf{H}\mathbf{R}_s\mathbf{H}^H + \sigma^2\mathbf{I}, \tag{8.5}$$

where $\mathbf{R}_s = E\{\mathbf{s}\mathbf{s}^H\}$ is the source covariance matrix and σ^2 the power of additive white noise. For uncorrelated sound sources, $\mathbf{R}_s$ is diagonal and (8.5) can be decomposed into three additive terms:

$$\mathbf{R} = \mathbf{h}_1\mathbf{h}_1^H S_1(f) + \mathbf{H}_{2:M}\mathbf{R}_{s_{2:M}}\mathbf{H}_{2:M}^H + \sigma^2\mathbf{I}, \tag{8.6}$$

where $S_1(f)$ is the power spectrum of the desired signal and $\mathbf{H}_{2:M} = [\mathbf{h}_2, ..\mathbf{h}_M]$. $\mathbf{R}_{s_{2:M}}$ can be obtained by removing the first row and the first column of $\mathbf{R}_s$. Substituting $\mathbf{w}^H\mathbf{h}_1 = 1$ in the MVDR objective function (8.4) yields,

$$\mathbf{w}^H\mathbf{R}\mathbf{w} = S_1(f) + \mathbf{w}^H\mathbf{H}_{21:M}\mathbf{R}_{s_{2:M}}\mathbf{H}_{2:M}^H\mathbf{w} + \sigma^2\mathbf{w}^H\mathbf{w} \tag{8.7}$$

By ignoring the first term, the MVDR optimization problem can be simplified to

$$\underset{\mathbf{w}}{\operatorname{argmin}} \ \mathbf{w}^H\mathbf{H}_{2:M}\mathbf{R}_{s_{2:M}}\mathbf{H}_{2:M}^H\mathbf{w} + \sigma^2\mathbf{w}^H\mathbf{w} \quad s.t \quad \mathbf{w}^H\mathbf{h}_1 = 1 \tag{8.8}$$

Note that (8.8) is independent of S_1. However, usually in reverberant rooms no knowledge about $\mathbf{h}_1$ is available in advance and it has to be estimated from the array data or approximated with steering vector $\mathbf{d}$ calculated from the main speaker location estimate. Since $\mathbf{h}_1$ includes both direct and indirect

paths, it can be decomposed into a sum of steering vector and echoes transfer function:

$$\mathbf{h}_1 = \mathbf{d} + \mathbf{h}_{echoes} \tag{8.9}$$

The attenuation factor has been intentionally dropped to avoid notational distraction. $\mathbf{R}$ can be rewritten as

$$\mathbf{R} = (\mathbf{h}_{echoes} + \mathbf{d})(\mathbf{h}_{echoes} + \mathbf{d})^H S_1(f) + \mathbf{H}_{2:M}\mathbf{R}_{s_{2:M}}\mathbf{H}_{2:M}^H + \sigma^2\mathbf{I} \tag{8.10}$$

By using $\mathbf{w}^H\mathbf{d} = 1$ as an approximation of the target transfer function $\mathbf{h}_1$, the objective function becomes

$$|\mathbf{w}^H\mathbf{h}_{echoes} + 1|^2 S_1(f) + \mathbf{w}^H\mathbf{H}_{2:M}\mathbf{R}_{s_{2:M}}\mathbf{H}_{2:M}^H\mathbf{w} + \sigma^2\mathbf{w}^H\mathbf{w} \tag{8.11}$$

Looking closely at this function reveals that the first term can be vanished if $\mathbf{w}^H\mathbf{h}_{echoes} = -1$ holds. Therefore the MVDR optimization problem tends to satisfy the $\mathbf{w}^H\mathbf{h}_{echoes} = -1$ constraint. However, satisfying this constraint results in removing the signal from the beamformer output. To clarify it, note that the signal component in the beamformer output can be written as $y_s = \mathbf{w}^H\mathbf{h}_1 s_1(f)$. Using (8.9) and the fact that the weight vector satisfies $\mathbf{w}^H\mathbf{d} = 1$ and $\mathbf{w}^H\mathbf{h}_{echoes} = -1$ constraints, the beamformer response to the signal transfer function $\mathbf{h}_1$ is $\mathbf{w}^H\mathbf{h}_1 = 0$ and thus $y_s = 0$. That is, by approximating the real transfer function (TF) with the steering vector $\mathbf{d}$, the MVDR beamformer tends to remove the desired signal. Actually, if we ignore the white noise term for sake of simplicity, by choosing $\mathbf{w}$ from $\mathbf{H}$ null space, the objective function reaches its minimum value, namely zero. It means $\mathbf{w}^H\mathbf{H} = 0$ and therefore $\mathbf{w}^H\mathbf{h}_1 = 0$.

A natural approach to address the signal cancellation problem is to estimate the acoustic transfer function and the microphone mismatches. This estimate is then used to achieve the signal-independent optimization problem in (8.8).

8.3 Transfer Function Estimation

In order to avoid signal cancellation, we aim to estimate $\mathbf{h}_1$ from the noisy signals $\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{n}$ received by microphones. This information can be extracted from the covariance matrix $\mathbf{R}$. Using spectral decomposition, $\mathbf{R}$ can be written as $\mathbf{R} = \mathbf{U}(\Lambda + \sigma^2\mathbf{I})\mathbf{U}^H$, where the first M columns of $\mathbf{U}$ are the orthonormal basis vectors of the signal and interference subspaces and σ^2 is

the white noise power which can be estimated as the average of the $N-M$ smallest eigenvalues of $\mathbf{R}$. Given the source position, the ideal steering vector can be simply calculated. However, because of acoustic characteristics of a room, the steering vector $\mathbf{d}$ may not lie in the subspace spanned by the $\mathbf{U}_{1:M}$ columns. One approach to estimate $\mathbf{h}_1$ is to find the closest vector to $\mathbf{d}$ which lie in the in subspace spanned by the first M columns of $\mathbf{U}$.

$$\underset{\mathbf{x}}{\operatorname{argmin}} \quad ||\mathbf{U}_{1:M}\mathbf{x} - \mathbf{d}||_2^F \tag{8.12}$$

Consequently, $\mathbf{h}_1 = \mathbf{U}_{1:M}\mathbf{x}$ and it belongs to the subspace $\mathbf{U}_{1:M}$. This is a projection problem and the solution can be written as $\mathbf{h}_1 = \mathbf{U}_{1:M}\mathbf{U}_{1:M}^H\mathbf{d}$, where $P = \mathbf{U}_{1:M}\mathbf{U}_{1:M}^H$ is the projection operator onto $\mathbf{U}_{1:M}$ subspace. However, although it can be an accurate guess when the steering vector and the real TF mismatch is fairly small, it may not work in more severe reverberant environments. To go one step further, we assume the availability of some additional information about the interferer locations. That is, direction of arrival of k interferers ($k \leq M$) can be derived from array data (which is reasonably simple at least for an imprecise estimate)[2]. Unlike some other methods, they do not necessarily need to be the k strongest sources. Defining $\mathbf{H}' = \mathbf{U}_{1:M}\Lambda_{M\times M}^{1/2}\mathbf{V}$ and thus $\mathbf{R} = \mathbf{H}'\mathbf{H}'^H$ one can infer that $\mathbf{H}'$ can be estimated up to an unknown multiplicative unitary matrix $\mathbf{V}$ from $\mathbf{R}$ decomposition. Our aim is to estimate this unitary matrix $\mathbf{V}$ and reconstruct $\mathbf{h}_1$ by $\hat{\mathbf{h}}_1 = \mathbf{U}_{1:M}\Lambda_{M\times M}^{1/2}\mathbf{v}_1$. Taking locations information into account, (8.12) can be extended as follows:

$$\underset{\mathbf{V},\Gamma}{\operatorname{argmin}} \quad ||\mathbf{U}_{1:M}\Lambda^{1/2}\mathbf{V}_{1:k}\Gamma - \mathbf{D}||_2^F \quad s.t \quad \mathbf{V}_{1:k}^H\mathbf{V}_{1:k} = \mathbf{I} \tag{8.13}$$

where Γ is a $k\times k$ diagonal weighting matrix which is necessary to model the unknown attenuation factor and $\mathbf{D} = [\mathbf{d}_1, ..., \mathbf{d}_k]$ is the steering matrix. It is worth to note that in case of $k = 1$, (8.13) reduces to the simple least squares problem in (8.12). However, unlike (8.12), there is no straightforward solution for it. The optimization problem in (8.13) is known as weighted orthogonal Procrustes problem (WOPP) and can be seen as a linear least squares problem defined on a Stiefel manifold. A Stiefel manifold, commonly denoted as $\mathcal{V}_{m,n}$, is the set of all matrices $\mathbf{V}_{M\times N}$ having orthonormal columns.

[2]Since many systems nowadays use both audio and video channels to ease the human-machine interaction, like Microsoft's Kinect-Xbox, the location information could also come from the vision channel.

Usually, solutions suggested for the optimization problem in 8.13 have two steps: Given Γ, they try to find the optimum $\mathbf{V}$ and use it in the second step to find the optimal Γ. This forms an iterative solution that converges to a local minimum. Here, we employ the algorithm suggested in [Dos10] with small modifications to work with complex matrices. The complete iterative channel matrix estimation algorithm is listed in Algorithm 8.1.

Algorithm 8.1: Iterative channel matrix estimation

Input: M eigenvectors $\mathbf{U}_{1:M}$, steering matrix $\mathbf{D}$ and eigenvalues Λ.
Set $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_k] = \Lambda^{1/2}\mathbf{U}_{1:M}^H\mathbf{D}$ and $\mathbf{V}_{1:k}^0 = \mathbf{I}$.

For $t = 1, \ldots,$ **do**

(a) Compute $\gamma_i = \dfrac{\mathbf{a}_i^H\mathbf{v}_i}{\mathbf{v}_i^H\Lambda\mathbf{v}_i}$ where $\mathbf{v}_i$ is i-th column of $\mathbf{V}^t$
and set $\Gamma = \text{diag}([\gamma_1, ..., \gamma_i])$

(b) Choose ρ such that $\rho\mathbf{I} - \Lambda$ is positive-definite.
Compute $\mathbf{c}_i = \gamma_i\mathbf{a}_i + |\gamma_i|^2(\rho\mathbf{I} - \Lambda)\mathbf{v}_i$ for $i = 1, ..., k$
and set $\mathbf{C} = [\mathbf{c}_1, ..., \mathbf{c}_k]$.
By orthogonal decomposition of $\mathbf{C}$ get $\mathbf{C} = \mathbf{U}_c\Lambda_c\mathbf{V}_c^H$
and set $\mathbf{V}^{t+1} = \mathbf{U}_c\mathbf{V}_c^H$

(c) $\Phi(t{+}1) = 2\sum_{i=1}^{k} real(\gamma_i\mathbf{v}_i^H\mathbf{a}_i) - \sum_{i=1}^{k}|\gamma_i|^2\mathbf{v}_i^H\Lambda\mathbf{v}_i$
Terminate if $\Phi(t{+}1) - \Phi(t) \approx 0$

End
Output: Γ and $\mathbf{V}^{t+1}$.

Simulations have shown that it converges in less than 20 iterations. This algorithm may be computationally expensive. However, as long as the transfer function of a room does not change rapidly, it only needs to be run infrequently. The output of this algorithm is an estimate of the channel matrix $\mathbf{H}$ which is used in (8.8) to achieve a signal-independent MVDR beamformer.

8.4 Simulation Results

The adaptive beamformer is implemented in the frequency domain with overlap-add 2048 point FFT filterbank and sampling frequency of $f_s = 44100$. The non-uniform linear array consists of 8 microphones as depicted in Figure 8.3. For evaluation of the proposed algorithm, two different simulation cases have been chosen.

Case I: In the first scenario, two speech signals in a reverberant room with $T_{60} = 100ms$ (T_{60} is the reverberation time [3]) have been assumed. The origin of the coordinate system is at the center of the microphone array. The desired speaker and the interferer are placed at coordinates $[0, 2.5m, 0]$ and $[3m, 2.5m, 0]$, respectively. A mismatch of less than 2 dB is assumed

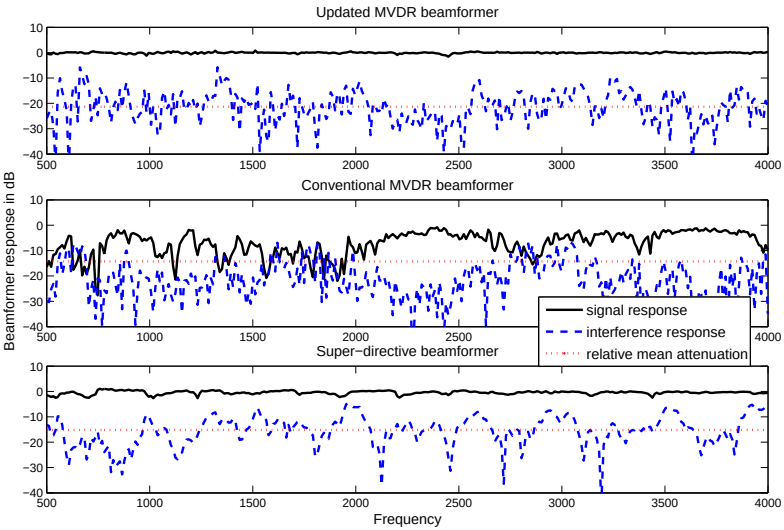

Figure 8.1: Proposed MVDR (top), conventional MVDR (middle) and super-directive beamformer responses to the signal and the interference.

[3] T_{60} is perceived as the time for the sound intensity to drop below -60 dB of the original sound after the sound source ceases.

between the microphone transfer functions. The beampattern can be defined as $BP(\mathbf{h}(f)) = |\mathbf{w}^H\mathbf{h}(f)|^2$ and beampattern values at $\mathbf{h} = \mathbf{h}_1$ (desired signal transfer function) and $\mathbf{h} = \mathbf{h}_2$ (interferer) can be interpreted as the beamformer responses to the desired signal and the interferer signal, respectively. These responses are shown in Figure 8.1 for the proposed, conventional MVDR and super-directive beamformers. Super-directive beamformers assume diffuse noise which is a very common model for reverberant environments. Note that beampattern at $\mathbf{h} = \mathbf{h}_1$ can be interpreted as $\frac{\text{SNR}_{\text{out}}}{\text{SNR}_{\text{in}}}$ where SNR is defined as signal to noise ratio and at $\mathbf{h} = \mathbf{h}_2$, $BP(\mathbf{h}_2(f)) = \frac{\text{INR}_{\text{out}}}{\text{INR}_{\text{in}}}$ with INR being interferer to noise ratio. According to these two equalities, the beampattern can be seen as a performance evaluation measure. The higher its value is at $\mathbf{h}_1$, the larger the SNR improvement and the lower its value is at $\mathbf{h}_2(f)$, the more the inference attenuation is.

The average of $\frac{BP(\mathbf{h}_2(f))}{BP(\mathbf{h}_1(f))}$ over all frequencies is called relative mean attenuation and is shown with a horizontal line for all beamformers in Figure 8.1. As it can be seen, the fixed super-directive beamformer can attenuate the interference up to 15 dB on average while the MVDR with updated constraint with our algorithm, can achieve 7 dB more interference reduction. Also, the beamformer response to $\mathbf{h}_1$ shows less fluctuations and thus less signal distortion for the updated MVDR than the super-directive beamformer. However, as expected, the conventional MVDR response to the target TF shows that it severely distorts the desired signal.

Case II: In the second scenario, the traditional and the proposed MVDR beamformer's performance in the presence of large target steering vector error have been studied. Figure 8.2 shows the beampattern of the MVDR beamformer with updated constraint (the proposed method) versus the traditional MVDR beamformer at 1 kHz. $-15\,^\circ$ of error in both interference and signal steering vectors has been assumed. Therefore both columns of the steering matrix in (13) are imprecise. Nevertheless, as it can be seen in Figure 8.2, the updated MVDR achieves both robustness against $15\,^\circ$ steering vector error and high interference reduction which is around 30 dB at interference direction. The solid line also reveals a slight shift in the null position (from $40\,^\circ$ to $33\,^\circ$ which leads to about 10 dB degradation in interference reduction (from 40 dB to 30 dB). More experiments have shown that the proposed algorithm is robust against even larger target direction errors at the expense of this shift in null position which can be seen as a trade-off between the noise reduction and robustness.

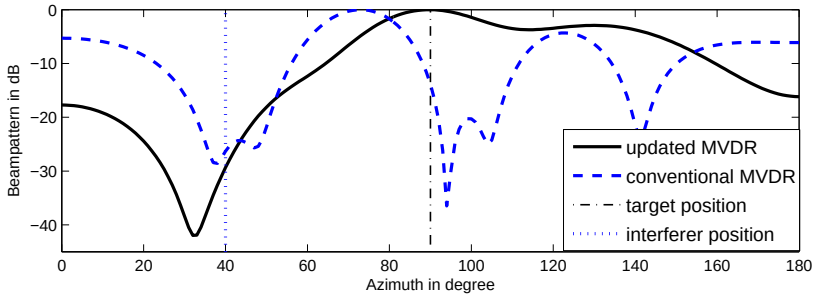

Figure 8.2: Beampattern of the updated constraint MVDR and the traditional MVDR at f=1000Hz. Vertical lines mark the source and the interferer positions. The signal vector as well the interferer vector are assumed to have an error of -15 degree.

The proposed algorithm prevents signal attenuation by shifting the main lobe to the correct azimuth while traditional MVDR performance dramatically worse. The main advantage of this method over other similar methods of robust constraint set design like [ZXG06] is that it does not widen the main lobe (since it results in spatial resolution reduction), but rather shifts it to the correct angle by extracting covariance matrix information.

8.5 Experiments with ETHDigits Dataset

Throughout this section, we used the ETHDigits dataset for evaluation of the proposed algorithms. ETHDigits is an audio-visual dataset recorded in a highly reverberant office room with T_{60} (reverberation time) longer than 1 second. The audio data was captured by 8 microphones that were arranged as in Figure 8.3 and RGB video signals with the resolution of 640x480 per frame were recorded by a Kinect for Xbox 360. Unlike the datasets used in the previous chapters, we did not record the visual data under controlled conditions. Depending on how many lights were on during a recording session, what time of day it was and whether the sun was shining, the amount of light in the room may largely vary, which in turn, results in large variation of the visual data quality. The ETHDigits dataset was recorded from 15 speakers and each speaker repeated a sequence of numbers from one to ten for 5 times (with no randomization). The numbers were presented on a computer screen

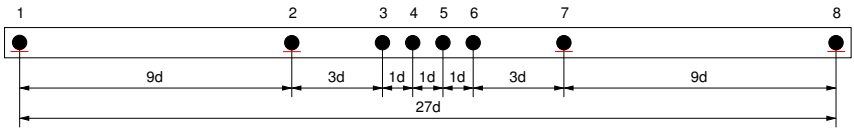

Figure 8.3: Linear microphone array with non-uniform spacing between microphones. d is 3 cm and the total length of the microphone array is 81 cm. The microphones 1, 2, 7 and 8 constitute a uniformly spaced four-channel microphone array used in some experiments.

located about 45 centimeters away from talkers and both Kinect and the microphone array were mounted on top of this screen.

In the first set of experiments, we evaluated the performance of the proposed beamforming algorithm (updated MVDR). Figure 8.4 demonstrates the performance of the audio only recognizer for each speaker when: (I) updated MVDR algorithm was applied to the output of our 8 channel microphone array to enhance the SNR of the audio signal fed to A-ASR, (II) only output of one microphone was passed to A-ASR (III) updated MVDR was applied to the output of a 4 channel microphone array and its output was then fed to A-ASR. The four channel microphone array was simply constructed by only considering the outputs of 4 microphones out of 8 microphones, as shown in Figure 8.3.

As expected, the update MVDR beamforming method with the 8-channel microphone array outperforms both the single-channel and the 4-channel system over all speakers, except speaker 15. Using the 8-channel microphone array yields 97.21%$\pm$1.41 average recognition accuracy over 15 speakers, compared with 95.38%$\pm$1.66 and 96.55%$\pm$1.60 of the single-channel and the 4-channel system, respectively. Figure 8.5 demonstrates the performance of visual speech recognizer when ISCN, MAPP and ISCN+MAPP (as discussed in Chapter 6) features are used to represent visual data. The average recognition accuracy of ISCN, MAPP and ISMA (ISCN+MAPP) based V-ASR over 15 speakers are 35.25%$\pm$4.24, 44.40%$\pm$2.98 and 46.88%$\pm$3.95, respectively. Similar to the Oulu, AVletter and GRID datasets in Chapter 6, the combination of ISCN and MAPP features, outperforms each of them individually, suggesting that ISCN and MAPP features are complimentary feature sets. In fact, it can be observed in Figure 8.5 that when ISCN and MAPP

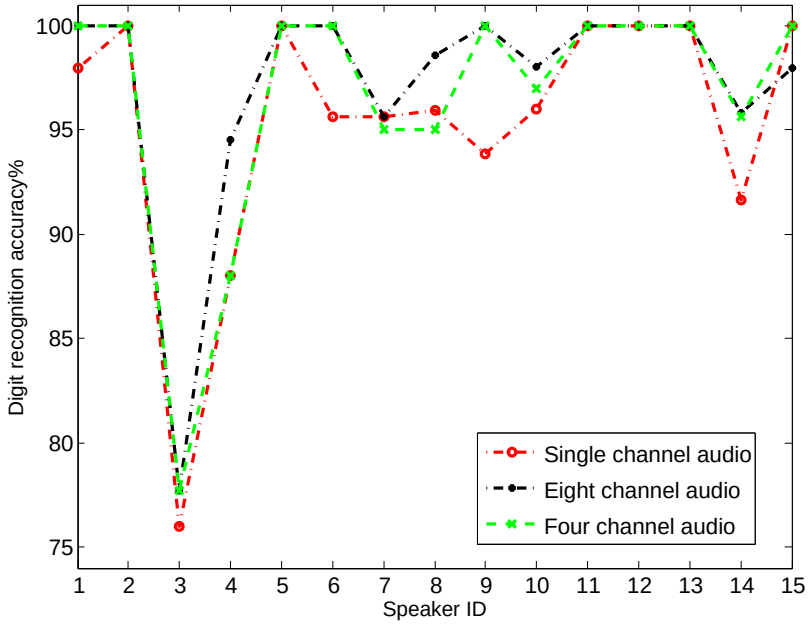

Figure 8.4: Recognition accuracy per speaker on ETHDigits for 8-channel, 4-channel and single-channel systems. In 8-channel and 4-channel systems updated MVDR are applied to enhance the the audio signal quality.

features obtain almost the same accuracy, their combination achieves a significantly higher recognition rate than each of them alone. For instance, V-ASR with ISMA features achieves about 65% accuracy on speaker 14 which is 15% higher than the performance of ISCN and MAPP, individually. The same trend can be seen for speakers 7, 9 and 11.

The recognition rate of the audio-visual recognizer for each speaker is shown in Figure 8.6. AV-ASR performance has been reported for two fusion strategies: (I) Audio and visual features (ISMA features were taken as visual features) had equal weights and (II) an individual weight was assigned to each feature of the audio-visual feature set. In this case the weights were estimated by means of the AUC based approach developed in Chapter 7. It

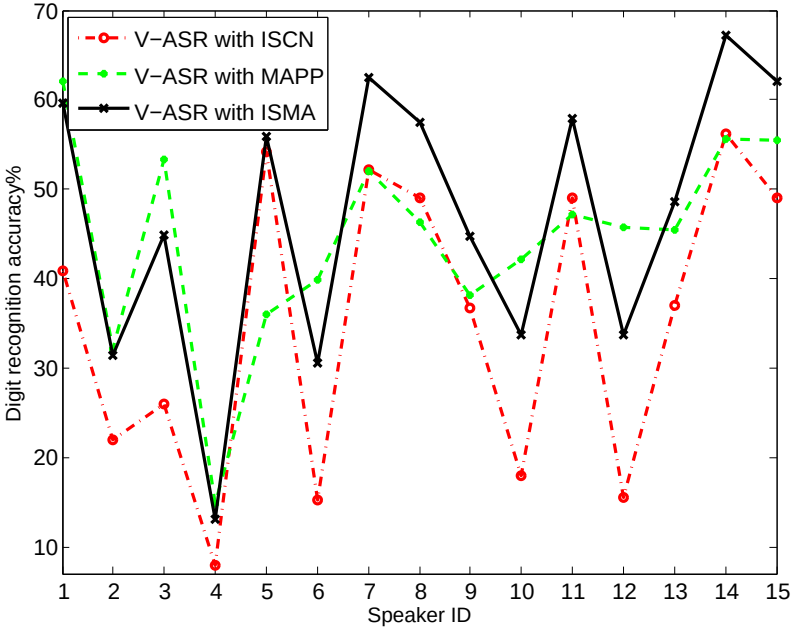

Figure 8.5: Recognition accuracy per speaker on ETHDigits for visual-only speech recognizer with ISCN, MAPP and ISCN+MAPP (ISMA) features.

is clear that weighted audio-visual fusion largely outperforms the simple uniform weight strategy. The weighted AV-ASR reaches 95.54%$\pm$1.5 accuracy compared with only 70.78%$\pm$4.51 recognition rate of uniform strategy. As it was already discussed in Chapter 7, using the one-weight per feature strategy increases the robustness of the system in mismatching training and test conditions by assigning higher weights to more robust features i.e., features with less variations over different speakers. By reducing the effect of unreliable features, AV-ASR can work reasonably accurate even when one modality completely fails. This, for instance, can be observed in the case of speaker 4 where visual-only recognizer can hardly perform better than random guessing while AV-ASR reaches 89% accuracy for this speaker.

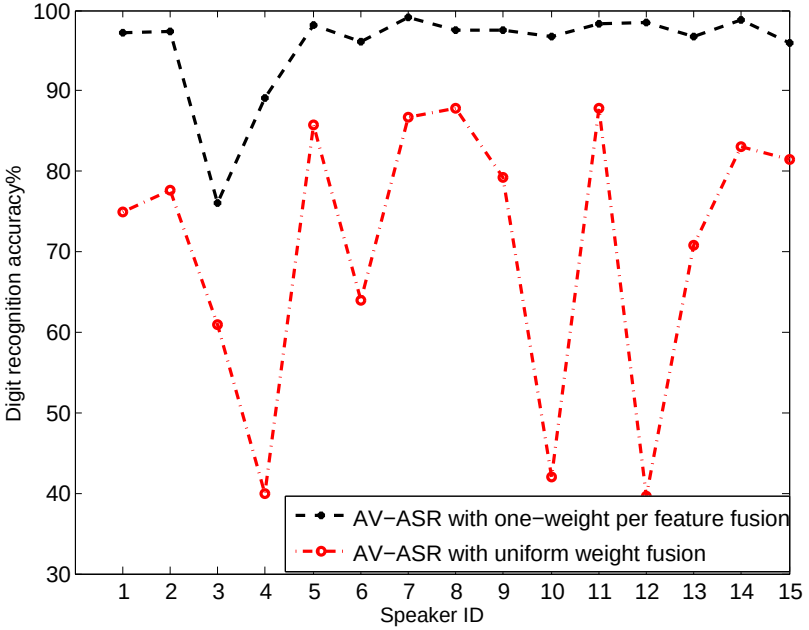

Figure 8.6: Recognition accuracy per speaker on ETHDigits for AV-ASR when all features have qaul weights (uniform weight strategy) and when one individual weight is assigned to each feature. These weights are estimated with the AUC based weight strategy presented in Algorithm 7.2.

8.6 Conclusion

In this chapter, we developed an algorithm to approximate the transfer function of the room from the covariance matrix of the multichannel audio data. It was shown that this problem can be formulated as an instance of the weighted Procrustes problem. This transfer function is then used to develop a more robust beamforming method against the signal cancellation problem. Using the proposed beamforming method yields 2% performance improvement of the audio-only speech recognizer when microphone array consists of 8 microphones and obtains 1.11% when 4-channel microphone array is utilized. The ETHDigits dataset which is an eight-channel audio-visual data is then used to

evaluate the AV-ASR system with a beamforming block in a highly reverberant office room with fairly low-quality visual data compared with Oulu and GRID datasets used in Chapter 6. The experiments showed that even though the visual speech recognizer in this adverse condition is unreliable, with a proper audio-visual fusion strategy AV-ASR can obtain 95.54% accuracy.

Chapter 9

Conclusion

9.1 Achievements

In this thesis we have developed several feature selection and boosting methods which can be used to construct robust audio-visual speech recognition and voice activity detection systems.

The main advantage of our proposed feature selection method, COBRA, is that it guarantees a non-zero lower bound on the normalized score[1] of selected feature set, which can be interpreted as a goodness measure of the selected feature set.

Since appearance based visual features are highly speaker dependent, we used ISCN feature extraction that generates translation- and scale-invariant features. Its drawback, however, is that it extracts tens of thousands of features from a relatively small image. Thus, we employed COBRA in order to reduce the ISCN feature vector dimensionality. Statistical models built upon the selected variables showed superior robustness against speaker variations and lighting conditions.

Boosting methods are based on the idea of creating a highly accurate classifier by combining many weak and inaccurate classifiers. It can be seen as a meta-algorithm that maintains a distribution over the sample space. At each iteration a weak hypothesis minimizing the weighted loss is learned and the

[1] normalized with respect to the score of the optimal feature set

weights (over the samples) are updated, accordingly. The output (strong hypothesis) is a convex combination of the weak hypotheses. Unlike most of the previously suggested boosting methods, in our boosting framework, MABoost, the booster has direct control over the weights, making it more suitable for boosting problems subject to some distribution constraints. We derived several theoretically and practically appealing algorithms (including SparseBoost, SmoothBoost, m-MABoost etc.) and more importantly provided some proof techniques that can be used to translate many other online learning algorithms into boosting methods.

By means of MABoost, we developed two practically important applications in speech processing: a reliable voice activity detector and a robust lip reading system. Our proposed voice activity detector can be trained in a semi-supervised manner, which is an important requirement in some applications. Our lip reading system is based on the generalization of the MABoost framework to multiclass setting. It is a decision tree based lip reading system which is highly suitable for the sparse features used in this system.

Finally, we devised an information fusion strategy based on AUC maximization. We showed that in this method, the weights assigned to likelihood values should be optimized with respect to a more robust criterion than a simple accuracy rate. We utilized AUC to estimate the likelihood weights and showed that under some simplifying assumptions this criterion can be seen as the expectation of accuracy rate with respect to uniformly distributed mismatch error. Therefore, maximizing it may result in a more robust system against mismatch conditions. Our experiments showed that this fusion strategy leads to a robust digit classification system with an accuracy rate favorably comparable with adaptive systems.

9.1.1 Perspectives

This thesis should be seen as a rather broad investigation for various robust visual features and machine learning techniques aiming to improve audio-visual speech recognition systems. Even though our current implementation of AV-ASR needs more refinements in order to be used in a commercial product, my inner Jules Verne believes in a close future there will be continuous AV-ASR based applications available on smart phones and intelligent vehicles which provide reasonable accuracy.

A realization of a reasonable lip reading system may at least satisfy the following requirements:

1. It should reasonably work in the speaker-independent mode.
2. It should be robust against illumination variations.
3. It can adapt itself to particular users.

and a successful AV-ASR should at least:

1. have a robust audio-visual utterance detector.
2. enjoy a robust audio-visual information fusion scheme, in order to cope with possible audio or visual modality failures.

The first two requirements were directly investigated and addressed in this thesis. We have shown that employing a pool of scale invariant features extracted from multiple space colors yields a high level of robustness against inter-speaker and illumination variations. Two important aspects of this approach are (I) Sparse representation of visual information (II) Constructing a strong classifier based on decision trees which can take advantage of feature vector sparsity. While the first property provides robustness against undesired variations, the second property guarantees to efficiently learn the underlying hypothesis. Due to these two properties, the proposed method can have further applications including visual emotion recognition, video classification and video search and scene detection.

While the third requirement has not been directly discussed in this thesis, most of the proposed algorithms can be easily modified to take speaker adaption into account. However, some of the theoretical results and performance guarantees could only be proven for batch learning setting and not for online learning which is a requirement in speaker adaption. Conducting further research on online boosting algorithms with PAC learning property may generalize our theoretical results to online learning settings.

Audio-visual voice activity detection is a prerequisite in many audio-visual applications. We proposed a robust audio-visual voice activity detector which can be trained in a semi-supervised manner. This interesting property can be achieved by noting the fact that both audio and visual signals represent the same underlying event: speech. Following the same line of reasoning, it is plausible to apply this semi-supervised training procedure to other audio-visual applications, including audio-visual speech recognition and audio-visual content search which are more complex than a simple AV-VAD. Each of these applications, however, may pose extra technical challenges which need to be addressed first. For instance, to apply this technique

to audio-visual phoneme recognition, we have to deal with the fact that an audio-based phoneme recognizer achieves much higher classification accuracy than its video-based counterpart. Thus, labeling the data by iterating over audio- and video-based classifiers may not converge to a meaningful result. One remedy to address this problem is to reduce the classes by clustering the phonemes into visems (or even broader speech units than visems) so that audio and video-based classifiers yield almost similar classification accuracy over them. Further works on this direction may result in a highly desirable (or highly scary!) autonomous artificial intelligence systems (perhaps mono-task systems) which can learn and improve on itself.

Finally, by means of the boosting framework and proof techniques suggested in Section 4 we solved two open problems. First, we showed that it is possible to select only a percentage of samples (in many datasets half of the samples or even less) at each round of training and still achieve 100% classification accuracy and second, we derive the first proof for Madaboost algorithm presented in [DW00].

9.1.2 Open Problems

Several new problems have also emerged from our work. Over the last two decades it was shown that non-convex loss functions may have some desirable properties such as robustness against labeling noise [LS10] and better scalability [CSWB06]. This raises the question that whether using a non-convex divergence function in MABoost can still result in a provable boosting algorithm?

The second question is related to the weights of the samples in a boosting algorithm. In common boosting methods, the weights constitute a probability distribution over the sample space. However, as shown in this work, this seems to not be a necessary condition. For instance in SparseBoost, the weights are projected onto a hypercube rather than a probability simplex. The main question is: under which conditions the boosting weights can be projected onto other convex sets than probability simplex, while the boosting algorithm still converges? and what are the characteristics of these convex sets?

Finding answers for these two questions may have big practical and theoretical impacts. It is known that using a non-convex loss function may lead to a boosting method which is robust against labeling noise. Moreover, projecting the weights onto a larger convex set than probability simplex, may enable

us to develop an algorithm that can learn the underlying hypothesis from an infinite amount of data in a very efficient manner (in fact exponentially fast) without using all the samples in the training data which would take an infinite amount of time. Given an algorithm satisfying these two properties, it is plausible to develop a semi-supervised audio-visual based algorithm that can be trained over unlabeled data gathered from the Internet.

Appendix A

Complementary Fundamentals

This appendix contains proofs of the lemmas and theorems presented in Chapter 4. Before proceeding with the proofs, some definitions and facts need to be reminded.

A.1 Definitions and Preliminaries

Definition A.1. Margin *Given a final hypothesis* $f(\mathbf{x}) = \sum_{t=1}^{T} \eta_t h_t(\mathbf{x})$*, the margin of a sample* $(\mathbf{x}_j, a_j)$ *is defined as* $m(\mathbf{x}_j) = a_j f(\mathbf{x}_j) / \sum_{t=1}^{T} \eta_t$*. Moreover, the margin of a set of examples denoted by* $m_{\mathcal{D}}$ *is the minimum of margins over the examples, i.e.,* $m_{\mathcal{D}} = \min_{\mathbf{x}} m(\mathbf{x}_j)$*.*

Lemma A.2. Duality between max-margin and min-edge *The minimum edge* γ_{min} *that can be achieved over all possible distributions of the training set is equal to the maximum margin (*$m^* = \max_{\eta} m_{\mathcal{D}}$*) of any linear combination of hypotheses from the hypotheses space.*

This lemma is discussed in details in [FS96a] and [RW05]. It is the direct result of von Neumann's minmax theorem and simply means that the maximum achievable margin is γ_{min}.

A.2 Proof of Theorem 4.7

The proof of the maximum margin property of MABoost is almost the same as the proof of Theorem 4.6.

Let's assume the i^{th} sample has the worst margin, i.e., $m_{\mathcal{D}} = m(\mathbf{x}_i)$. Let all entries of the error vector $\mathbf{w}^*$ to be zero except its i^{th} entry which is set to be 1. Following the same approach as in Theorem 4.6, (see equation (4.13)), we get

$$\sum_{t=1}^{T} \mathbf{w}^{*\top} \eta_t \mathbf{d}_t - \mathbf{w}_t^{\top} \eta_t \mathbf{d}_t \leq \sum_{t=1}^{T} \frac{1}{2} \eta_t^2 ||\mathbf{d}_t||_*^2 + B_{\mathcal{R}}(\mathbf{w}^*, \mathbf{w}_1) - B_{\mathcal{R}}(\mathbf{w}^*, \mathbf{w}_{T+1}) \tag{A.1}$$

With our choice of $\mathbf{w}^*$ it is easy to verify that the first term on the left side of the inequality is $m_{\mathcal{D}} \sum_{t=1}^{T} \eta_t = -\sum_{t=1}^{T} \mathbf{w}^{*\top} \eta_t \mathbf{d}_t$. By setting $C = B_{\mathcal{R}}(\mathbf{w}^*, \mathbf{w}_1)$, ignoring the last term $B_{\mathcal{R}}(\mathbf{w}^*, \mathbf{w}_{T+1})$, replacing $||\mathbf{d}_t||_*^2$ with its upper bound L and using the identity $\sum_{t=1}^{T} \mathbf{w}_t^{\top} \eta_t \mathbf{d}_t = -\sum_{t=1}^{T} \eta_t \gamma_t$ the above inequality is simplified to

$$-m_{\mathcal{D}} \sum_{t=1}^{T} \eta_t \leq L \sum_{t=1}^{T} \frac{1}{2} \eta_t^2 - \sum_{t=1}^{T} \eta_t \gamma_t + C \tag{A.2}$$

Replacing η_t with the value suggested in Theorem 4.7, i.e., $\eta_t = \dfrac{\gamma_t}{L\sqrt{t}}$ and dividing both sides by $\sum_{t=1}^{T} \eta_t$, gives

$$\frac{\sum_{t=1}^{T} (\frac{1}{\sqrt{t}} - \frac{1}{t}) \gamma_t^2}{\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \gamma_t} - \frac{LC}{\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \gamma_t} \leq m_{\mathcal{D}} \tag{A.3}$$

The first term is minimized when $\gamma_t = \gamma_{\min}$. Similarly to the first term, the second term is maximized when γ_t is replaced by its minimum value. This gives the following lower bound for $m_{\mathcal{D}}$:

$$\gamma_{\min} \frac{\sum_{t=1}^{T} \frac{1}{\sqrt{t}} - \frac{1}{t}}{\sum_{t=1}^{T} \frac{1}{\sqrt{t}}} - \frac{LC}{\gamma_{\min} \sum_{t=1}^{T} \frac{1}{\sqrt{t}}} \leq m_{\mathcal{D}} \tag{A.4}$$

Considering the facts that $\int_1^{T+1} \frac{dx}{\sqrt{x}} \leq \sum_{t=1}^{T} \frac{1}{\sqrt{t}}$ and $1 + \int_1^T \frac{dx}{x} \geq \sum_{t=1}^{T} \frac{1}{t}$, we get

$$\gamma_{\min} - \frac{1+\log T}{2\sqrt{T+1}-2}\gamma_{\min} - \frac{LC}{\gamma_{\min}(\sqrt{T+1}-1)} \leq m_{\mathcal{D}} \tag{A.5}$$

Now by taking $\nu = \frac{1+\log T}{2\sqrt{T+1}-2}\gamma_{\min} + \frac{LC}{\gamma_{\min}(\sqrt{T+1}-1)}$, we have $\gamma_{\min} - \nu \leq \gamma_{\min}$. It is clear from (A.5) that ν approaches zero as T tends to infinity with a convergence rate proportional to $\frac{\log T}{\sqrt{T}}$. It is noteworthy that this convergence rate is slightly worse than that of TotalBoost which is $O(\frac{1}{\sqrt{T}})$.

A.3 Proof of Lemma 4.8

Remember that $\hat{\Pi}_{\mathcal{S}}(\mathbf{y}) = \Pi_{\mathcal{S}}\Big(\Pi_{\mathcal{K}}(\mathbf{y})\Big)$. Our goal is to show that $B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) \geq B_{\mathcal{R}}\big(\mathbf{x}, \hat{\Pi}_{\mathcal{S}}(\mathbf{y})\big)$.

To this end, we only need to repeatedly apply Lemma 4.2, as follows

$$B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) \geq B_{\mathcal{R}}\big(\mathbf{x}, \Pi_{\mathcal{K}}(\mathbf{y})\big) \tag{A.6}$$

$$B_{\mathcal{R}}\big(\mathbf{x}, \Pi_{\mathcal{K}}(\mathbf{y})\big) \geq B_{\mathcal{R}}\big(\mathbf{x}, \hat{\Pi}_{\mathcal{S}}(\mathbf{y})\big) \tag{A.7}$$

which completes the proof.

A.4 Proof of the Boosting Algorithm for Combined Datasets

We have to show that when the convex set is defined as

$$\mathcal{S}_c = \{\mathbf{w} | \sum_{i=1}^{N} w_i = 1, 0 \leq w_i \ \ \forall i \in \mathcal{A} \ \ \wedge \ \ 0 \leq w_i \leq \frac{k}{N} \ \ \forall i \in \mathcal{B}\} \tag{A.8}$$

the error of the final hypothesis on $\mathcal{A}$, i.e., $\epsilon_{\mathcal{A}}$, converges to zero while the error on $\mathcal{B}$ is guaranteed to be $\epsilon_{\mathcal{B}} \leq \frac{1}{k}$.

First, we show the convergence of $\epsilon_{\mathcal{A}}$ to zero. This is easily obtained by setting $\mathbf{w}^*$ to be an error vector with zero weights over the training samples from $\mathcal{B}$ and $\frac{1}{\epsilon_{\mathcal{A}} N_{\mathcal{A}}}$ weights over the training set $\mathcal{A}$. One can verify that $w^* \in$

$\mathcal{S}_c$, thus the proof of Theorem 4.6 holds and subsequently, the error bounds in (4.8) stating that $\epsilon_{\mathcal{A}} \to 0$ as the number of iterations increases.

To show the second part of the theorem that is $\epsilon_{\mathcal{B}} \leq \frac{1}{k}$, vector $\mathbf{w}^*$ is selected to be an error vector with zero weights over the training samples from $\mathcal{A}$ and $\frac{1}{\epsilon_{\mathcal{B}} N_{\mathcal{B}}}$ weights over the training set $\mathcal{B}$. Note that, as long as $\epsilon_{\mathcal{B}}$ is greater than $\frac{1}{k}$, this $w^* \in \mathcal{S}_c$. Thus, for all $\frac{1}{k} \leq \epsilon_{\mathcal{B}}$ the proof of Theorem 4.6 holds and as the bounds in (4.8) show, the error decreases as the number of iterations increases. In particular in a finite number of rounds, the classification error on $\mathcal{B}$ reduces to $\frac{1}{k}$ which completes the proof.

A.5 Proof of Theorem 4.9

We use proof techniques similar to those given in [DSST10], with a slight change to take the normalization step into account.

By replacing $\mathbf{z}_{t+1}$ in the projection step from the update step, the projection step can be rewritten as

$$\mathbf{y}_{t+1} = \arg\min_{\mathbf{y} \in \mathcal{R}_+} \frac{1}{2}||\mathbf{y} - \mathbf{y}_t|| - \eta_t \mathbf{y}^\top \mathbf{d}_t + \alpha_t \eta_t ||\mathbf{y}||_1 \tag{A.9}$$

This optimization problem can be highly simplified by noting that the variables are not coupled. Thus, each coordinate can be independently optimized. In other words, it can be decoupled into N independent 1-dimensional optimization problems.

$$y^i_{t+1} = \arg\min_{0 \leq y_i} \frac{1}{2}||y_i - y^i_t|| - \eta_t y_i d^i_t + \alpha_t \eta_t y_i \tag{A.10}$$

The solution of (A.10) can be written as

$$y^i_{t+1} = \max(0, y^i_t + \eta_t d^i_t - \alpha_t \eta_t) \tag{A.11}$$

This simple solution gives a very efficient and simple implementation for SparseBoost. From (A.10) it is clear that for $d^i_t < 0$ (i.e., when i^{th} sample is classified correctly), $-\eta_t y d^i_t$ acts as the ℓ_1 norm regularization and pushes y^i_{t+1} towards zero while $\alpha_t \eta_t$ enhance sparsity by pushing all weights to zero.

Let $\mathbf{w}^*$ to be the same error vector as defined in Theorem 4.6. We start this proof by again deriving the progress bounds on each step of the algorithm.

The optimality of $\mathbf{y}_{t+1}$ for (A.9) implies that

$$(\mathbf{w}^* - \mathbf{y}_{t+1})^\top(-\eta_t \mathbf{d}_t + \alpha_t \eta_t r'(\mathbf{y}) + \mathbf{y}_{t+1} - \mathbf{y}_t) \geq 0 \tag{A.12}$$

where $r'(\mathbf{y})$ is a sub-gradient vector of the ℓ_1 norm function $r(\mathbf{y}) = \sum_{i=1}^{N} y_i$. Moreover, due to the convexity of $r(\mathbf{y})$, we have

$$\alpha_t \eta_t r(\mathbf{y}_{t+1})^\top(\mathbf{w}^* - \mathbf{y}_{t+1}) \leq \alpha_t \eta_t \big(r(\mathbf{w}^*) - r(\mathbf{y}_{t+1})\big) \tag{A.13}$$

We thus have

$$\begin{aligned}
&(\mathbf{w}^* - \mathbf{y}_t)^\top \eta_t \mathbf{d}_t + \alpha_t \eta_t \big(r(\mathbf{y}_{t+1}) - r(\mathbf{w}^*)\big) \\
&\leq (\mathbf{w}^* - \mathbf{y}_t)^\top \eta_t \mathbf{d}_t + \alpha_t \eta_t (\mathbf{y}_{t+1} - \mathbf{w}^*)^\top r'(\mathbf{y}_{t+1}) \\
&= (\mathbf{w}^* - \mathbf{y}_{t+1})^\top \eta_t \mathbf{d}_t + \alpha_t \eta_t (\mathbf{y}_{t+1} - \mathbf{w}^*)^\top r'(\mathbf{y}_{t+1}) + (\mathbf{y}_{t+1} - \mathbf{y}_t)^\top \eta_t \mathbf{d}_t \\
&= (\mathbf{w}^* - \mathbf{y}_{t+1})^\top (\eta_t \mathbf{d}_t - \alpha_t \eta_t r'(\mathbf{y}_{t+1}) - \mathbf{y}_{t+1} + \mathbf{y}_t) \\
&+ (\mathbf{w}^* - \mathbf{y}_{t+1})^\top (\mathbf{y}_{t+1} - \mathbf{y}_t) + (\mathbf{y}_{t+1} - \mathbf{y}_t)^\top \eta_t \mathbf{d}_t
\end{aligned} \tag{A.14}$$

where the first inequality follows (A.13). Now, from the optimality condition in (A.12), the first term in the last equation is non-positive and thus, can be ignored.

$$\begin{aligned}
&(\mathbf{w}^* - \mathbf{y}_t)^\top \eta_t \mathbf{d}_t + \alpha_t \eta_t \big(r(\mathbf{y}_{t+1}) - r(\mathbf{w}^*)\big) \\
&\leq (\mathbf{w}^* - \mathbf{y}_{t+1})^\top (\mathbf{y}_{t+1} - \mathbf{y}_t) + (\mathbf{y}_{t+1} - \mathbf{y}_t)^\top \eta_t \mathbf{d}_t \\
&= \frac{1}{2}||\mathbf{w}^* - \mathbf{y}_t||_2^2 - \frac{1}{2}||\mathbf{y}_{t+1} - \mathbf{y}_t||_2^2 - \frac{1}{2}||\mathbf{w}^* - \mathbf{y}_{t+1}||_2^2 + (\mathbf{y}_{t+1} - \mathbf{y}_t)^\top \eta_t \mathbf{d}_t \\
&\leq \frac{1}{2}||\mathbf{w}^* - \mathbf{y}_t||_2^2 - \frac{1}{2}||\mathbf{y}_{t+1} - \mathbf{y}_t||_2^2 \\
&\quad - \frac{1}{2}||\mathbf{w}^* - \mathbf{y}_{t+1}||_2^2 + \frac{1}{2}||\mathbf{y}_{t+1} - \mathbf{y}_t||_2^2 + \frac{1}{2}\eta_t^2 ||\mathbf{d}_t||_*^2
\end{aligned} \tag{A.15}$$

where the first equation follows from Lemma 4.3 (or direct algebraic expansion in this case) and the second inequality from Lemma 4.5.

By summing the left and right sides of the inequality from 1 to T, replacing $||\mathbf{d}_t||_*^2$ with its upperbound N and substituting 1 for $r(\mathbf{w}^*)$, we get

$$\begin{aligned}
\sum_{t=1}^{T} \mathbf{w}^{*\top} \eta_t \mathbf{d}_t \leq \sum_{t=1}^{T} \mathbf{y}_t^\top \eta_t \mathbf{d}_t + \sum_{t=1}^{T} \frac{N}{2}\eta_t^2 + \frac{1}{2}||\mathbf{w}^* - \mathbf{y}_1||_2^2 \\
+ \sum_{t=1}^{T} \alpha_t \eta_t \big(1 - r(\mathbf{y}_{t+1})\big)
\end{aligned} \tag{A.16}$$

Now, replacing $r(\mathbf{y}_{t+1})$ with its lower bound, i.e, 0 and using the fact that $\sum_{t=1}^{T} \mathbf{w}^{*\top} \eta_t \mathbf{d}_t \geq 0$ (as shown in (4.14)) and $\sum_{t=1}^{T} \mathbf{y}_t^{\top} \eta_t \mathbf{d}_t = -\sum_{t=1}^{T} \eta_t \gamma_t ||\mathbf{y}_t||_1$, yields

$$0 \leq -\sum_{t=1}^{T} \eta_t \gamma_t ||\mathbf{y}_t||_1 + \sum_{t=1}^{T} \frac{N}{2} \eta_t^2 + \frac{1}{2} ||\mathbf{w}^* - \mathbf{y}_1||_2^2 + \sum_{t=1}^{T} \alpha_t \eta_t \qquad \text{(A.17)}$$

Taking derivative w.r.t η_t and setting it to zero, gives the optimal η_t as follows

$$\eta_t = \frac{\gamma_t ||\mathbf{y}_t||_1 - \alpha_t}{N} \qquad \text{(A.18)}$$

This equation implies that α_t should be smaller than $\gamma_t ||\mathbf{y}_t||_1$ or otherwise η_t becomes smaller than zero. Setting $\alpha_t = (1-k)\gamma_t ||\mathbf{y}_t||_1$ where k is a constant smaller than or equal to 1, results in $\eta_t = \frac{k}{N} \gamma_t ||\mathbf{y}_t||_1$. Replacing this value for η_t in (A.17) and noting that $\frac{1}{2} ||\mathbf{w}^* - \mathbf{y}_1||_2^2 = \frac{1-\epsilon}{2N\epsilon}$ gives the following bound on the training error

$$\epsilon \leq \frac{1}{1 + c \sum_{t=1}^{T} \gamma_t^2 ||\mathbf{y}_t||_1^2} \qquad \text{(A.19)}$$

where $c = \frac{1}{k^2}$ is a constant factor depending on the choice of α_t. To prove that ϵ approaches zero as T increases, we still have to provide an evidence that $\sum_{t=1}^{T} \gamma_t^2 ||\mathbf{y}_t||_1^2$ is a divergent series. There are different possibilities to approach this problem. Here, we show that in the case of $\alpha_t = 0$, the ℓ_1 norm of weights $||\mathbf{y}_t||_1$ can be bounded away from zero (i.e., $||\mathbf{y}_t||_1 \geq C > 0$) and thus, $\sum_{t=1}^{T} \gamma_t^2 ||\mathbf{y}_t||_1^2 \geq T \gamma_{\min}^2 C^2$.

To this end, we rewrite y_t^i from (A.11) as

$$\begin{aligned} y_t^i &= \max(0, y_{t-1}^i + \eta_{t-1} d_{t-1}^i - \alpha_{t-1} \eta_{t-1}) \\ &\geq y_{t-1}^i + \eta_{t-1} d_{t-1}^i - \alpha_{t-1} \eta_{t-1} \\ &\geq \frac{1}{N} + \sum_{t'=1}^{t-1} \eta_{t'} d_{t'} - \sum_{t'=1}^{t-1} \alpha_{t'} \eta_{t'} \end{aligned} \qquad \text{(A.20)}$$

where the last inequality is achieved by recursively applying the first inequality to y_{t-1}^i. At any arbitrary round t, either the algorithm has already converged and $\epsilon = 0$ or there is at least one sample that is classified wrongly by the ensemble classifier $H_t(\mathbf{x}) = \sum_{l=1}^{t} \eta_l h_l(\mathbf{x})$. Now, without loss of

generality, assume the i^{th} sample is wrongly classified at round t. That is, $\sum_{t'=1}^{t-1} \eta_{t'} d_{t'} > 0$ (look at (4.14)). Now, for $\alpha_t = 0$, the weight of the wrongly classified sample i is

$$y_t^i \geq \frac{1}{N} + \sum_{t'=1}^{t-1} \eta_{t'} d_{t'} \geq \frac{1}{N} \tag{A.21}$$

That is, $||\mathbf{y}_t||_1 \geq \frac{1}{N}$. This gives a lousy (but sufficient for our purpose) lower bound on $||\mathbf{y}_t||_1$. Replacing $||\mathbf{y}_t||_1$ with its lower bound $\frac{1}{N}$ in (A.19), yields

$$\epsilon \leq \frac{N^2}{1 + T\gamma^2} \tag{A.22}$$

where γ is the minimum edge over all γ_t.

A.6 Proof of Entropy Projection onto Hypercube

Lemma A.3. *Let $\mathcal{R}(\mathbf{w}) = \sum_{i=1}^N w_i \log w_i - w_i$. Then the Bregman projection of a positive vector $\mathbf{z} \in \mathbb{R}_+^N$ onto the unit hypercube $\mathcal{K} = [0,1]^N$ is $y_i = \min(1, z_i), i = 1, \ldots, N$.*

To show the correctness of the above lemma, i.e., that the solution of the Bregman projection

$$\mathbf{y} = \arg\min_{\mathbf{y} \in \mathcal{K}} B_{\mathcal{R}}(\mathbf{y}, \mathbf{z}) \tag{A.23}$$

is $y_i = \min(1, z_i)$, we only need to show that $\mathbf{y}$ satisfies the optimality condition

$$(\mathbf{v} - \mathbf{y})^\top \nabla B_{\mathcal{R}}(\mathbf{y}, \mathbf{z}) \geq 0 \quad \forall \mathbf{v} \in \mathcal{K} \tag{A.24}$$

Given $\mathcal{R}(\mathbf{w}) = \sum_{i=1}^N w_i \log w_i - w_i$, the gradient of $B_{\mathcal{R}}$ is

$$\nabla B_{\mathcal{R}}(\mathbf{y}, \mathbf{z}) = \sum_{i=1}^T \log \frac{y_i}{z_i} \tag{A.25}$$

Hence,

$$(\mathbf{v} - \mathbf{y})^\top \nabla B_{\mathcal{R}}(\mathbf{y}, \mathbf{z}) = \sum_{i \in \{i : z_i \geq 1\}} (v_i - y_i) \log \frac{y_i}{z_i} + \sum_{i \in \{i : z_i < 1\}} (v_i - y_i) \log \frac{y_i}{z_i} \tag{A.26}$$

For $z_i \geq 1$, y_i is equal to 1. That is, $\log \frac{y_i}{z_i} = \log \frac{1}{z_i} < 0$. On the other hand, since $v_i \leq 1$, $(v_i - y_i) = (v_i - 1) \leq 0$. Thus, the first sum in (A.26) is always non-negative. The second sum is always zero since $\log \frac{y_i}{z_i} = \log 1 = 0$. That is, the optimality condition (A.26) is non-negative for all $\mathbf{v}$ which completes the proof.

A.7 Proof of Theorem 4.11

Its proof is essentially the same as the proof of the lazy version of MABoost with a few differences. Before proceeding further, some definitions and facts should be re-emphasized.

First of all, since $\mathcal{R}(\mathbf{w}) = \sum_{i=1}^{N} w_i \log w_i - w_i$ is $\frac{1}{N}$-strongly convex (see [Sha12, p. 136]) with respect to ℓ_1 norm (and not 1-strongly as in Theorem 4.6), the following inequality holds for the Bregman divergence:

$$B_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) \geq \frac{1}{2N} ||\mathbf{x} - \mathbf{y}||_1^2 \tag{A.27}$$

Moreover, the following lemma which bounds $||\mathbf{y}_t||$ is essential for our proof.

Lemma A.4. *For all t, $||\mathbf{y}_t||_1 \geq N\epsilon_t$ where ϵ_t is the error of the ensemble hypothesis $H_t(\mathbf{x}) = \sum_{l=1}^{t} \eta_l h_l(\mathbf{x})$ at round t.*

This lemma holds due to the fact that

$$y_t^i = \min(1, z_t^i) = \min(1, e^{\sum_{l=1}^{t} \eta_l d_l^i}) = \min(1, e^{-a_i H_t(\mathbf{x}_i)}) \tag{A.28}$$

where $H_t(\mathbf{x}) = \sum_{l=1}^{t} \eta_l h_l(\mathbf{x})$ is the output of the algorithm at round t. If $H_t(\mathbf{x}_i)$ makes a mistake on classifying $\mathbf{x}_i$, $-a_i H_t(\mathbf{x}_i)$ will be greater than zero and thus, $y_t^i = 1$. For the samples that are classified correctly, $-a_i H_t(\mathbf{x}_i) \leq 0$ and thus, $0 \leq y_t^i \leq 1$. That is, $N\epsilon_t =$ number of wrongly classified samples at round $t \leq \sum_{i=1}^{N} y_t^i = ||\mathbf{y}_t||_1$.

We are now ready to proceed with the proof of Theorem 4.11. Let $\mathbf{w}^* = [w_1^*, \cdots, w_N^*]^\top$ to be a vector where $w_i^* = 1$ if $f(\mathbf{x}_i) \neq a_i$, and 0 otherwise. Similar to the proof of the lazy update, we are going to bound the $\sum_{t=1}^{T} (\mathbf{w}^* - \mathbf{y}_t)^\top \eta_t \mathbf{d}_t$.

$$
\begin{aligned}
(\mathbf{w}^* - \mathbf{y}_t)^\top \eta_t \mathbf{d}_t &= (\mathbf{y}_{t+1} - \mathbf{y}_t)^\top \big(\nabla\mathcal{R}(\mathbf{z}_{t+1}) - \nabla\mathcal{R}(\mathbf{z}_t)\big) \\
&+ (\mathbf{z}_{t+1} - \mathbf{y}_{t+1})^\top \big(\nabla\mathcal{R}(\mathbf{z}_{t+1}) - \nabla\mathcal{R}(\mathbf{z}_t)\big) \\
&+ (\mathbf{w}^* - \mathbf{z}_{t+1})^\top \big(\nabla\mathcal{R}(\mathbf{z}_{t+1}) - \nabla\mathcal{R}(\mathbf{z}_t)\big) \\
&\leq \frac{1}{2N}||\mathbf{y}_{t+1} - \mathbf{y}_t||^2 + \frac{N}{2}\eta_t^2||\mathbf{d}_t||_*^2 + B_\mathcal{R}(\mathbf{y}_{t+1}, \mathbf{z}_{t+1}) \\
&- B_\mathcal{R}(\mathbf{y}_{t+1}, \mathbf{z}_t) + B_\mathcal{R}(\mathbf{z}_{t+1}, \mathbf{z}_t) \\
&- B_\mathcal{R}(\mathbf{w}^*, \mathbf{z}_{t+1}) + B_\mathcal{R}(\mathbf{w}^*, \mathbf{z}_t) - B_\mathcal{R}(\mathbf{z}_{t+1}, \mathbf{z}_t) \\
&\leq \frac{1}{2N}||\mathbf{y}_{t+1} - \mathbf{y}_t||^2 + \frac{N}{2}\eta_t^2||\mathbf{d}_t||_*^2 - B_\mathcal{R}(\mathbf{y}_{t+1}, \mathbf{y}_t) \\
&+ B_\mathcal{R}(\mathbf{y}_{t+1}, \mathbf{z}_{t+1}) - B_\mathcal{R}(\mathbf{y}_t, \mathbf{z}_t) - B_\mathcal{R}(\mathbf{w}^*, \mathbf{z}_{t+1}) + B_\mathcal{R}(\mathbf{w}^*, \mathbf{z}_t)
\end{aligned} \tag{A.29}
$$

where the first inequality follows from applying Lemma 4.5 to the first term and Lemma 4.3 to the rest of the terms and the second inequality is the result of applying the exact version of Lemma 4.2 to $B_\mathcal{R}(\mathbf{y}_{t+1}, \mathbf{z}_t)$. Moreover, according to inequality (A.27) $B_\mathcal{R}(\mathbf{y}_{t+1}, \mathbf{y}_t) - \frac{1}{2N}||\mathbf{y}_{t+1} - \mathbf{y}_t||^2 \geq 0$ and hence these terms can be ignored in (A.29). Summing up the inequality (A.29) from $t = 1$ to T, yields:

$$
-B_\mathcal{R}(\mathbf{w}^*, \mathbf{z}_1) \leq \sum_{t=1}^{T} \frac{N}{2}\eta_t^2 - \sum_{t=1}^{T} \eta_t\gamma_t||\mathbf{y}_t||_1 \tag{A.30}
$$

It is important to remark that $||\mathbf{y}_t||_1$ appearing in the last term is due to the fact that $\mathbf{w}_t = \frac{\mathbf{y}_t}{||\mathbf{y}_t||_1}$ and thus, $\mathbf{y}_t^\top \eta_t \mathbf{d}_t = \mathbf{w}_t^\top \eta_t \mathbf{d}_t ||\mathbf{y}_t||_1 = \eta_t\gamma_t||\mathbf{y}_t||_1$.

Now, by replacing $\eta_t = \epsilon_t\gamma_t$ in the above equation and noting that $B_\mathcal{R}(\mathbf{w}^*, \mathbf{z}_1) = N - N\epsilon$, we get:

$$
-N(1-\epsilon) \leq \sum_{t=1}^{T} \frac{N}{2}\epsilon_t^2\gamma_t^2 - \sum_{t=1}^{T} \epsilon_t\gamma_t^2||\mathbf{y}_t||_1 \tag{A.31}
$$

From Lemma A.4, it is evident that $||\mathbf{y}_t||_1 \geq N\epsilon_t$. Moreover, since $\epsilon \leq \epsilon_t$, it can be replaced by ϵ, as well (though very pessimistic). As usual, γ_t is also replaced with the min edge, denoted by γ. Applying these lower bounds to the equation (A.31), yields

$$
\epsilon^2 \leq \frac{2(1-\epsilon)}{T\gamma^2} \leq \frac{1}{T\gamma^2} \tag{A.32}
$$

which indicates that the proposed version of MadaBoost needs at most $O(\frac{1}{\epsilon^2\gamma^2})$ iterations to converge.

A.8 Multiclass Weak-learning Condition

In this appendix, we show that the weak-learning condition adopted for Mu-MABoost is the optimal weak-learning condition in the sense that it is the weakest condition that can be assumed while guaranteeing the boosting property. To this end, we first show that the weak-learning condition in Definition 3 is equivalent to that of ADABoost.MR.

As before, assume all the samples belong to class 0. Define $\mathcal{W}$ to be the collection of $N \times K$ weight matrices $\mathbf{W}$ satisfying the following conditions:

1. $W_{i,j} \geq 0$ for $j \neq 0$
2. $W_{i,0} = -\sum_{j\neq 0} W_{i,j}$

Further, consider the matrix $\mathbf{1}_h$ to be an $N \times K$ matrix whose $(i,j)^{\text{th}}$ entry is 1 if $h(\mathbf{x}_i = j)$. The following equation then describes the ADABoost.MR weak-learning condition presented in [MS13].

$$\forall\, \mathbf{W} \in \mathcal{W}, \exists h \in \mathcal{H} : \mathbf{W} \bullet \mathbf{1}_h \leq 0 \tag{A.33}$$

It is straightforward to show that the weak-learning condition in (A.33) is equivalent to that of in Definition 3. Let *Corr* be the set of indices of the correctly classified samples by hypothesis h. By expanding the left side of inequality (A.33), we get

$$\begin{aligned}\mathbf{W} \bullet \mathbf{1}_h = \sum_{n=1}^{N} W_{i,h(\mathbf{x}_i)} &= \sum_{i\in Corr} W_{i,0} + \sum_{i\notin Corr} W_{i,j} \\ &= \sum_{i\notin Corr} W_{i,j} - \sum_{i\in \text{Corr}} \sum_{j\neq 0} W_{i,j}\end{aligned} \tag{A.34}$$

The right side of the last equation in (A.34) is equal to $\mathbf{W} \bullet \mathbf{D}$ in Definition 3. Thus, both ADABoost.MR and Mu-MABoost adopt the same weak-learning condition. Moreover, according to Theorem 6 in [MS13], this condition is the minimal weak-learning condition that can be adopted such that the boosting algorithms tends the training error to zero.

Bibliography

[AB96] D. W. Aha and R. L. Bankert. A comparative evaluation of sequential feature selection algorithms. In *Learning from Data: Artificial Intelligence and Statistics V*. Springer-Verlag, 1996.

[Abr63] N. Abramson. *Information Theory and Coding*. McGraw-Hill, New York, 1963.

[AF06] A. E. Abdel-Hakim and A. A. Farag. Csift: A sift descriptor with color invariant characteristics. In *Proceedings of CVPR*, 2006.

[AITT00] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama. Greedily finding a dense subgraph. *Journal of Algorithms*, 2000.

[AM08] I. Almajai and B. Milner. Using audio-visual features for robust voice activity detection in clean and noisy speech. In *Proceedings of EUSPC*, 2008.

[AMM+07] M. Aoki, K. Masuda, H. Matsuda, T. Takiguchi, and Y. Ariki. Voice activity detection by lip shape tracking using EBGM. In *Proceedings of the 15th International Conference on Multimedia*. ACM, 2007.

[Bat94] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Trans. on Neural Networks*, 5:537–550, 1994.

[BGL02] N. H. Bshouty, D. Gavinsky, and M. Long. On boosting with polynomially bounded distributions. *Journal of Machine Learning Research*, 2002.

[BLM01] S. Ben-David, P. Long, and Y. Mansour. Agnostic boosting. In *COLT*. 2001.

[BM98] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of COLT*, 1998.

[BM12] J. Bruna and S. Mallat. Invariant scattering convolution networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2012.

[Bog07] V. I. Bogachev. *Measure Theory*. Springer, 2007.

[BP10] K. S. Balagani and V. V. Phoha. On the feature selection criterion based on an approximation of multidimensional mutual information. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2010.

[BPZL12] G. Brown, A. Pocock, M. Zhao, and M. Luján. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *Journal of Machine Learning*, 2012.

[Bre97] L. Breiman. Pasting bites together for prediction in large data sets and on-line. Technical report, Dept. Statistics, Univ. California, Berkeley, 1997.

[Bre99] L. Breiman. Prediction games and arcing algorithms. *Neural Computation*, 1999.

[Bre01] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[Bro09] G. Brown. A new perspective for information theoretic feature selection. In *Proceedings of Artificial Intelligence and Statistics*, 2009.

[BS08] J. K. Bradley and R. E. Schapire. Filterboost: Regression and classification on large datasets. In *NIPS*. 2008.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[Cas79] K. R. Castleman. *Digital Image Processing*. Prentice Hall Professional Technical Reference, 1st edition, 1979.

[CBCS06] M. Cooke, J. Barker, S. Cunningham, and X. Shao. An audio-visual corpus for speech perception and automatic speech recognition. *The Journal of the Acoustical Society of America*, 2006.

[CBL06] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[CDM02] C.C. Chibelushi, F. Deravi, and J. S. Mason. A review of speech-based bimodal recognition. *IEEE Trans. on Multimedia*, 2002.

[CET01] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[CH97] G. I. Chiou and J. Hwang. Lipreading from color video. *IEEE Trans. on Image Processing*, 1997.

[CH23] L. Cappelletta and H. Harte. Phoneme-to-viseme mapping for visual speech recognition. In *In Proceedings of Patter Recognition Applications and Methods*, 2023.

[Che01] T. Chen. Audiovisual speech processing. *IEEE Signal Processing Magazine*, 2001.

[CHLN08] S. Cox, R. Harvey, Y. Lan, and J. Newman. The challenge of multispeaker lip-reading. *In International Conference on Auditory-Visual Speech Processing*, 2008.

[CM93] M. Cohen and D. Massaro. Modeling coarticulation in synthetic visual speech. In *Models and Techniques in Computer Animation*. 1993.

[CM03] C. Cortes and M. Mohri. AUC optimization vs. error rate minimization. In *NIPS*. MIT Press, 2003.

[CR98] T. Chen and R. R. Rao. Audio-visual integration in multimodal communication. *Proceedings of the IEEE*, 1998.

[CSO10] P. M. Ciarelli, E. O. T. Salles, and E. Oliveira. An evolving system based on probabilistic neural network. In *Proceedings of BSNN*, 2010.

[CSWB06] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Trading convexity for scalability. In *Proceedings of ICML*, 2006.

[CT91] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, NY, USA, 1991.

[DGFVG12] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool. Real-time facial feature detection using conditional regression forests. In *CVPR*, 2012.

[DL00] S. Dupont and J. Luettin. Audio-visual speech modeling for continuous speech recognition. *IEEE Trans. on Multimedia*, 2000.

[Doa92] J. Doak. An evaluation of feature selection methods and their application to computer security. Technical Report CSE-92-18, University of California at Davis, 1992.

[Dos10] M. B. Dosse. Anisotropic orthogonal Procrustes analysis. *Journal of Classification*, 27, 2010.

[DSST10] J. C. Duchi, S. Shalev-shwartz, Y. Singer, and A. Tewari. Composite objective mirror descent. In *COLT*, 2010.

[DW00] C. Domingo and O. Watanabe. Madaboost: A modification of AdaBoost. In *COLT*, 2000.

[DYXY07] W. Dai, Q. Yang, G. Xue, and Y. Yong. Boosting for transfer learning. In *ICML*, 2007.

[FA10] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[Fan61] R. Fano. *Transmission of Information: A Statistical Theory of Communications*. The MIT Press, Cambridge, MA, 1961.

[FDV12] B. Frénay, G. Doquire, and M. Verleysen. On the potential inadequacy of mutual information for feature selection. In *Proceedings of ESANN*, 2012.

[FF97] J. H. Friedman and U. Fayyad. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1997.

[FHF11] P. Flach, J. Hernández-orallo, and C. and Ferri. A coherent interpretation of AUC as a measure of aggregated classification performance. In *Proceedings of ICML*, 2011.

[FHT98] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 1998.

[Fre95] Y. Freund. Boosting a weak learning algorithm by majority. *Journal of Information and Computation*, 1995.

[FS96a] Y. Freund and R. E. Schapire. Experiments with a New Boosting Algorithm. In *ICML*, 1996.

[FS96b] Y. Freund and R. E. Schapire. Game theory, on-line prediction and boosting. In *COLT*, 1996.

[FS97] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 1997.

[FT74] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. on Computers*, 1974.

[FY83] A. M. Frieze and J. Yadegar. On the quadratic assignment problem. *Discrete Applied Mathematics*, 1983.

[Gav03] D. Gavinsky. Optimally-smooth adaptive boosting and application to agnostic learning. *Journal of Machine Learning Research*, 2003.

[GBW01] S. Gannot, D. Burshtein, and E. Weinstein. Signal enhancement using beamforming and nonstationarity with applications to speech. *IEEE Trans. on Signal Processing*, 49:1614 –1626, 2001.

[GCSP13] Z. Golumbic, G. B. Cogan, C. E. Schroeder, and D. Poeppel. Visual input enhances selective speech envelope tracking in auditory cortex at a "cocktail party". 2013.

[GPP+12] L. Grippo, L. Palagi, M. Piacentini, V. Piccialli, and G. Rinaldi. SpeeDP: an algorithm to compute SDP bounds for very large max-cut instances. *Mathematical Programming*, 2012.

[GSM13] F. G. German, D. L. Sun, and G. J. Mysore. Towards a practical lipreading system. In *Proceedings of CVPRInterspeech*, 2013.

[Gur09] M. Gurban. *Multimodal feature extraction and fusion for audio-visual speech recognition*. PhD thesis, 4292, STI, EPF Lausanne, 2009.

[GVN+01] H. Glotin, D. Vergyr, C. Neti, G. Potamianos, and J. Luettin. Weighting schemes for audio-visual fusion in speech recognition. In *Proceedings of ICASSP*, 2001.

[GW95] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 1995.

[Han80] T. S. Han. Multiple mutual informations and multiple interactions in frequency data. *Information and Control*, 1980.

[Hat06] K. Hatano. Smooth boosting using an information-based criterion. In *Algorithmic Learning Theory*. 2006.

[Haz09] E. Hazan. A survey: The convex optimization approach to regret minimization. Working draft, 2009.

[HDY+12] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 2012.

[HES00] H. Hermansky, D. P. W. Ellis, and S. Sharma. Tandem connectionist feature extraction for conventional hmm systems. In *Proceedings of ICASSP*, 2000.

[Hie93] J. L. Hieronymus. Ascii phonetic symbols for the world's languages worldbet. *International Phonetic Association*, 93.

[HR70] M. Hellman and J. Raviv. Probability of error, equivocation and the Chernoff bound. *IEEE Trans. on Information Theory*, 1970.

[HSH99] O. Hoshuyama, A. Sugiyama, and A. Hirano. A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters. *IEEE Trans. on Signal Processing*, 47:2677–2684, 1999.

[HT01] D. J. Hand and R. J. Till. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Journal of Machine Learning Research*, 2001.

[HT09] K. Hatano and E. Takimoto. Linear programming boosting by column and row generation. In *Proceedings of Discovery Science*, 2009.

[Hu11] B. G. Hu. What are the differences between Bayesian classifiers and mutual-information classifiers? *IEEE Trans. on Neural Networks and Learning Systems*, 2011.

[HW99] M. Hollander and D. A. Wolfe. *Nonparametric Statistical Methods, 2nd Edition*. Wiley-Interscience, 1999.

[JB71] J. Jeffers, , and M. Barley. *Speechreading (Lipreading)*. Charles C Thomas Pub Ltd., 1971.

[JHL97] B. Juang, W. Hou, and C. Lee. Minimum classification error rate methods for speech recognition. *IEEE Trans. on Speech and Audio Processing*, 1997.

[KC02] N. Kwak and C. Choi. Input feature selection for classification problems. *IEEE Trans. on Neural Networks*, 2002.

[KHDM98] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1998.

[KK09] A. Kalai and V. Kanade. Potential-based agnostic boosting. In *NIPS*. 2009.

[KKG07] B. J. Killian, J. Y. Kravitz, and M. K. Gilson. Extraction of configurational entropy from molecular simulations via an expansion approximation. *Journal of Chemical Physics*, 2007.

[KMS08] A. Klaser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *proceedings of BMVC*, 2008.

[Koh96] R. Kohavi. *Wrappers for Performance Enhancement and Oblivious Decision Graphs*. PhD thesis, Stanford, CA, USA, 1996.

[KR13] T. Kinnunen and P. Rajan. A practical, self-adaptive voice activity detector for speaker verification with noisy telephone and microphone data. In *ICASSP*, 2013.

[KS01] D. Karger and N. Srebro. Learning Markov networks: Maximum bounded tree-width graphs. *Proceedings of the 12th Annual Symposium on Discrete Algorithms*, pages 392–401, 2001.

[KSS92] M. J. Kearns, R. E. Schapire, and L. M. Sellie. Toward efficient agnostic learning. In *COLT*, 1992.

[Lan00] T. Lane. Extensions of ROC analysis to multi-class domains. In *ICML-2000 Workshop on Cost-Sensitive Learning*, 2000.

[LCSC05] S. Lucey, T. Chen, S. Sridharan, and V. Chandran. Integration strategies for audio-visual speech processing: applied to text-dependent speaker recognition. *IEEE Trans. on Multimedia*, 2005.

[LHT+09] Y. Lan, R. Harvey, B. J. Theobald, E. Ong, and R. Bowden. Comparing visual features for lipreading. In *Auditory-Visual Speech Processing*, 2009.

[Lof04] J. Lofberg. YALMIP: a toolbox for modeling and optimization in MATLAB. In *Proceedings of Computer Aided Control Systems Design*, 2004.

[Low04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.

[LS10] P. M. Long and R. A. Servedio. Random classification noise defeats all convex potential boosters. *Journal of Machine Learning Research*, 2010.

[LT97] J. Luettin and N. Thacker. Speechreading using probabilistic models. *Computer Vision and Image Understanding*, 1997.

[LTB96] J. Luettin, N. A. Thacker, and S. Beet. Speechreading using shape and intensity information. In *International Conf. Spoken Language Proc.*, 1996.

[LTH+10] Y. Lan, B. J. Theobald, R. Harvey, E. J. Ong, and R. Bowden. Improving visual features for lipreading. In *Proceedings of AVSP*, 2010.

[MB06] P. Meyer and G. Bontempi. *On the Use of Variable Complementarity for Feature Selection in Cancer Classification*. Springer, 2006.

[MBBF99] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent. In *NIPS*, 1999.

[MCB+02] I. Matthews, T.F. Cootes, J.A. Bangham, S. Cox, and R. Harvey. Extraction of visual features for lipreading. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2002.

[McG54] W. McGill. Multivariate information transmission. *Trans. of the IRE Professional Group on Information Theory*, 1954.

[MHL+07] E. McDermott, T. J. Hazen, J. Le-Roux, A. Nakamura, and S. Katagiri. Discriminative training for large-vocabulary speech recognition using minimum classification error. *IEEE Trans. on Audio, Speech, and Language Processing*, 2007.

[MLS+13] V. P. Minotto, C. B. O. Lopes, J. Scharcanski, C. R. Jung, and B. Lee. Audiovisual voice activity detection based on microphone arrays and color information. *IEEE Journal of Selected Topics in Signal Processing*, 2013.

[MM02] Y. Mansour and D. McAllester. Boosting using branching programs. *Journal of Computer and System Sciences*, 2002.

[MS02] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of Computer Vision*, 2002.

[MS13] I. Mukherjee and R. E. Schapire. A theory of multiclass boosting. *Journal of Machine Learning Research*, 2013.

[Nag14] T. Naghibi. MABoost: An R package for classification, 2014. URL http://cran.r-project.org/web/packages/maboost.

[NC09] J. L. Newman and S. J. Cox. Automatic visual-only language identification: A preliminary study. In *Proceedings of ICASSP*, 2009.

[NF77] P. M. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Trans. on Computers*, 1977.

[NHP13] T. Naghibi, S. Hoffmann, and B. Pfister. Convex approximation of the NP-hard search problem in feature subset selection. In *Proceedings of ICASSP*, 2013.

[NSS04] J. Neumann, C. Schnörr, and G. Steidl. SVM-based feature selection by direct objective minimisation. In *Proceedings of DAGM*, 2004.

[NWF08] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 2008.

[ONS99] S. Okawa, T. Nakajima, and K. Shirai. A recombination strategy for multi-band speech recognition based on mutual information criterion. In *Proceedings of Eurospeech*, 1999.

[OPM02] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2002.

[PD00] F. Provost and P. Domingos. Well-trained PETs: Improving probability estimation trees, 2000. CDER Working Paper, Stern School of Business, New York University, NY.

[PG98] G. Potamianos and H. P. Graf. Discriminative training of HMM stream exponents for audio-visual speech recognition. In *Proceedings of ICASSP*, 1998.

[PGC08] S. Pachoud, S. Gong, and A. Cavallaro. Macro-cuboid based probabilistic matching for lip-reading digits. *Proceedings of CVPR*, 2008.

[PGTG02] E. K. Patterson, S. Gurbuz, Z. Tufekci, and J. N. Gowdy. CUAVE: A new audio-visual database for multimodal human-computer interface research. In *In Proc. of ICASSP*, 2002.

[PLD05] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2005.

[PLHZ04] H. Pan, S. E. Levinson, T. S. Huang, and L. Zhi-pei. A fused hidden Markov model with application to bimodal speech processing. *IEEE Trans. on Signal Processing*, 2004.

[PNLM04] G. Potamianos, C. Neti, J. Luettin, and I. Matthews. Audio–visual automatic speech recognition: an overview. *Visual and Audio–Visual Speech Processing*, 2004.

[PRW95] S. Poljak, F. Rendl, and H. Wolkowicz. A recipe for semidefinite relaxation for (0,1)-quadratic programming. *Journal of Global Optimization*, 1995.

[PSSD06] A. Potamianos, E. Sanchez-Soto, and K. Daoudi. Stream weight computation for multi-stream classifiers. In *Proceedings of ICASSP*, 2006.

[QV95] F. Qian and B. Van Veen. Quadratically constrained adaptive beamforming for coherent signals and interference. *IEEE Trans. on Signal Processing*, 43:1890–1900, 1995.

[QWP11] L. Qingju, W. Wenwu, and J. Philip. A visual voice activity detection method with adaboosting. In *Sensor Signal Processing for Defence, SSPD*, 2011.

[Rag88] P. Raghavan. Probabilistic construction of deterministic algorithms: approximating packing integer programs. *Journal of Computer and System. Sciences*, 1988.

[RCB97] M. G. Rahim, L. Chin-Hui, and J. Biing-Hwang. Discriminative utterance verification for connected digits recognition. *IEEE Trans. on Speech and Audio Processing*, 1997.

[Rez61] F. M. Reza. *An Introduction to Information Theory*. Dover Publications, Inc., New York, 1961.

[RHEC10] I. Rodriguez-Lujan, R. Huerta, Ch. Elkan, and C. S. Cruz. Quadratic programming feature selection. *Journal of Machine Learning Research*, 2010.

[RJ93] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[RS12] R. Rajavel and P. S. Sathidevi. Adaptive reliability measure and optimum integration weight for decision fusion audio-visual speech recognition. *Journal of Signal Processing Systems*, 2012.

[RW05] G. Rätsch and M. Warmuth. Efficient margin maximization with boosting. *Journal of Machine Learning Research*, 2005.

[SAS07] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional SIFT descriptor and its application to action recognition. In *Proceedings of Multimedia*, 2007.

[Sch90] R. E. Schapire. The strength of weak learnability. *Journal of Machine Learning Research*, 1990.

[Ser03] R. A. Servedio. Smooth boosting and learning with malicious noise. *Journal of Machine Learning Research*, 2003.

[Sha12] S. Shalev-Shwartz. Online learning and online convex optimization. *Found. Trends Machine Learning*, 2012.

[SK85] T. J. Shan and T. Kailath. Adaptive beamforming for coherent signals and interference. *IEEE Trans. on ASSP*, vol. 33:527–536, 1985.

[SKS99] J. Sohn, N. S Kim, and W. Sung. A statistical model-based voice activity detection. *IEEE Signal Processing Letters*, 6(1):1–3, 1999.

[SLS+05] K. Saenko, K. Livescu, M. Siracusa, K. Wilson, J. Glass, and T. Darrell. Visual speech recognition with loosely synchronized feature streams. In *Proceedings of ICCV*, 2005.

[SS98] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *COLT*, 1998.

[SS06] F. Sha and L. K. Saul. Large margin Gaussian mixture modeling for phonetic classification and recognition. In *Proceedings of ICASSP 2006.*, 2006.

[SS08] S. Shalev-Shwartz and Y. Singer. On the equivalence of weak learnability and linear separability: new relaxations and efficient boosting algorithms. In *COLT*, 2008.

[SW98] A. Srivastav and K. Wolf. Finding dense subgraphs with semidefinite programming. In *Approximation Algorithms for Combinatiorial Optimization*. Springer, 1998.

[SZ03] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *Proceedings of Computer Vision*, 2003.

[TA+97] T. M. Therneau, E. J. Atkinson, et al. An introduction to recursive partitioning using the RPART routines. Technical report, URL http://www.mayo.edu/hsr/techrpt/61.pdf, 1997.

[VD93] H. Vafaie and K. De-Jong. Robust feature selection algorithms. In *Proceedings of TAI*, 1993.

[vdSGS10] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2010.

[Vik06] T. Viklands. *Algorithms for the Weighted Orthogonal Procrustes Problem and other Least Squares Problems*. PhD thesis, Umeå University, Umeå, Sweden, 2006.

[vK70] J. von Kries. Influence of adaptation on the effects produced by luminous stimuli. *In MacAdam, D.L. (Ed.), Sources of Color Vision*, 1970.

[vL07] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 2007.

[WLR06] M. K. Warmuth, J. Liao, and G. Rätsch. Totally corrective boosting algorithms that maximize the margin. In *ICML*, 2006.

[WSC99] T. Wark, S. Sridharan, and V. Chandran. Robust speaker verification via fusion of speech and lip modalities. In *Proceeding of ICASSP*, 1999.

[YEG+06] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland. *The HTK Book, version 3.4*. Cambridge University Engineering Department, Cambridge, UK, 2006.

[Yeu91] R. W. Yeung. A new outlook on Shannon's information measures. *IEEE Trans. on Information Theory*, 1991.

[YNO09] T. Yoshida, K. Nakadai, and H. G. Okuno. Automatic speech recognition improved by two-layered audio-visual integration for robot audition. In *9th IEEE-RAS International conference on Humanoid Robots*, 2009.

[YYDS11] D. Ying, Y. Yan, J. Dang, and F. K. Soong. Voice activity detection based on an unsupervised learning framework. *IEEE Transactions on ASLP*, 2011.

[ZBP09] G. Zhao, M. Barnard, and M. Pietikainen. Lipreading with local spatiotemporal descriptors. *IEEE Trans. on Multimedia*, 2009.

[ZCCW13] N. A. Zaidi, J. Cerquides, M. J. Carman, and G. I. Webb. Alleviating naive Bayes attribute independence assumption by attribute weighting. *Journal of Machine Learning Research*, 2013.

[Zin03] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, 2003.

[ZJYH11] P. Zhao, R. Jin, T. Yang, and S. Hoi, C. Online AUC maximization. In *Proceedings of ICML*, New York, NY, USA, 2011. ACM.

[ZST10] X. Zhao, D. Sun, and K. Toh. A Newton-CG augmented Lagrangian method for semidefinite programming. *SIAM J. on Optimization*, 2010.

[ZXG06] Y. Zheng, P. Xie, and S. Grant. Robustness and distance discrimination of adaptive near field beamformers. *Proceedings of ICASSP*, 12:478–488, 2006.

[ZZHP14] Z. Zhou, G. Zhao, X. Hong, and M. Pietikäinen. A review of recent advances in visual speech decoding. *Image and Vision Computing*, 2014.

[ZZP11] Z. Zhou, G. Zhao, and M. Pietikainen. Towards a practical lipreading system. In *Proceedings of CVPR*, 2011.

[ZZRH09] J. Zhu, H. Zou, S. Rosset, and T. Hastie. Multi-class AdaBoost. *Statistics and its interface*, 2009.

Curriculum Vitae

1983	Born in Tehran, Iran
1990-2001	Primary and high school in Tehran
2001-2006	Studies in electrical engineering at University of Tehran
2006	Bs.c in electrical engineering
2006-2009	Master studies in telecommunication at Sharif university of Technology, Tehran
2009-2015	Research assistant and PhD student at the Speech Processing Group, Computer Engineering and Networks Laboratory, ETH Zürich