ASR-DRIVEN BINARY MASK ESTIMATION FOR ROBUST AUTOMATIC SPEECH RECOGNITION

DISSERTATION

Presented in Partial Fulfillment of the Requirements for

the Degree Doctor of Philosophy in the

Graduate School of The Ohio State University

By

William Hartmann, B.S., M.S.

Graduate Program in Computer Science & Engineering

The Ohio State University

2013

Dissertation Committee:

Prof. Eric Fosler-Lussier, Adviser

Prof. DeLiang Wang

Prof. Mary Beckman

ABSTRACT

Additive noise has long been an issue for robust automatic speech recognition (ASR) systems. One approach to noise robustness is the removal of noise information through segregation by binary time-frequency masks; each time-frequency unit in a spectro-temporal representation of the speech signal is labeled either noise-dominant or signal-dominant. The noise-dominant units are masked and their energy is removed from the signal. The ideal binary mask, computed given oracle information regarding the speech and noise sources, has been shown to provide significant improvements in speech intelligibility for humans. In this work, we investigate both methods of incorporating binary masks in ASR and methods for estimating the binary mask.

While applying binary masks to separation tasks for humans has been straightforward, the incorporation of binary masks in ASR has proved more difficult. The field of missing data ASR proposes methods to compensate for the effects of binary masks on cepstral feature calculation. We demonstrate, contrary to previous work, the direct use of the ideal binary mask performs at least as well as several missing data techniques when the acoustic features have been variance normalized.

Typical methods for binary mask estimation focus on low level acoustic features and little work attempts to incorporate higher level linguistic information in the estimation process. We propose an alternative masking criterion that forces the use of higher level information called the ASR-driven binary mask. The mask is defined by force aligning the true

word sequence with the acoustic model the of ASR system. Given oracle information, the ideal binary mask does perform better, but the ASR-driven binary mask may be easier to estimate. In addition, it easily allows for the incorporation of higher-level information in the estimation process.

We present a method for estimating the ASR-driven binary mask using a discriminatively trained sequence model. The hypotheses generated by a first pass through a recognition system are used to estimate the mask. Our proposed method significantly outperforms several methods for estimating the ideal binary mask. We also outline how the system could be used to iteratively improve the estimation.

Dedicated to my parents.

ACKNOWLEDGMENTS

As with all large projects, I could not have completed this thesis without the help and support of a large number of people.

I owe my knowledge to the many great professors I have had over the years. Of course, I must first thank my advisor, Professor Eric Fosler-Lussier; he happily took in a second year graduate student with no direction and instructed him in speech recognition and the use of semicolons. He was always supportive even though I likely gave him little reason to be early on. I am grateful for his desire to see his students succeed for their own benefit and for being the antithesis of the advisor portrayed in PhD Comics. I would also like to thank Professor Jim Davis for his support and guidance during my first year, Dr. Donna Byron for her support and trusting me to handle a project nearly unsupervised, Professor Mary Beckman for her instruction in linguistics and her detailed feedback from both my candidacy exam and thesis defense, and Professor DeLiang Wang for his instruction in all things related to the brain and hearing and his research guidance. I must also thank my undergraduate research advisor, Professor Richard Fox; without his help and instruction I would not have attended graduate school. He has continued to serve as an advisor and mentor and has been the one person I could always turn to for support and advice in all matters academic.

Without the support and friendship of my first labmates in the Computer Vision Lab, I may not have survived my first year. I would like to thank Alex Morison for spending

my first summer teaching me how to run cables through the ceiling to install cameras and motion detectors, Mark Keck for his great friendship, Karthik Sankaranarayanan for his cricket instruction, and Ambrish Tyagi and Vinay Sharma for teaching me about Indian culture and every word of Hindi that cannot be repeated in polite company.

My fellow SLATE lab members assistance and encouragement was invaluable; I will miss the weekly lab lunches. I would like to thank Jeremy Morris for his help in so many areas and his daily discussions about current events, Ilana Heintz for her linguistic expertise and her HTK webpage that I made use of long before I realized the author sat two desks away, Tim Weale for being the only person willing to discuss sports and for his unwavering pessimism, reminding me that things could always be worse, Rohit Prabhavalkar for taking over Tim's role when he graduated and for the many useful discussions about machine learning, Preethi Jyothi for always offering a handshake to congratulate me for every life event, Preethi Raghavan, with Rohit and Preethi #1, for telling me that everything my previous labmates taught me about India was wrong and for never agreeing on the answer to even the simplest question about India, Yanzhang He for his early morning company before the arrival of the other lab members, and Yi Ma for always making me laugh with his absurd stories. My graduate school experience would have been lonely and far less entertaining without them.

The members of the PNL lab, John Woodruff, Xiaojia Zhao, Arun Narayanan, Ke Hu, and Kun Han, provided valuable help with signal processing and source separation questions. I would especially like to thank John Woodruff for serving as my traveling companion and introducing me to my wife. I am grateful for the members of the CSE Administrative Staff, Catrena Collins, Tamera Cramer, Don Havard, Lynn Lyons, and Carrie Stein, for

always happily helping graduate students navigate the many partially documented requirements of a large university.

I thank my parents and brother for their love and support and, most importantly, for never questioning why I was still in school. When surrounded by so many intelligent people, it is easy to lose confidence in your own abilities; I appreciate that I could always rely on the confidence of my friends and family even when I questioned my own. I thank David Seiwert and his late father Don for helping me become the man I am today. Finally, I would like to thank my wife Yangqiao. She has taught me much about life and love and has encouraged me to do things I would have never done on my own. I appreciate the sacrifices she has made while I completed my PhD; hopefully it was worth the wait.

I acknowledge the financial support from The Ohio State University, the NSF GK-12 program, NSF grant IIS-0835396, and NSF CAREER grant IIS-0643901.

VITA

January 20, 1983	Born in Cincinnati, OH, USA
May, 2006	B.S., Computer Science Northern Kentucky University, Highland Heights, KY, USA
June, 2010	M.S., Computer Science The Ohio State University, Columbus, OH, USA

PUBLICATIONS

Journal Articles

R. Fox and W. Hartmann, "Using Context to Improve Hand-Written Character Recognition," *in the International Society for Advanced Science and Technology Transactions on Computers and Intelligent Systems*, Vol. 1, No. 1, pp. 40–49, 2009.

Conference Papers and Books

W. Hartmann and E. Fosler-Lussier, ASR-Driven Top-Down Binary Mask Estimation Using Spectral Priors, *in Proceedings of IEEE ICASSP*, (Kyoto, Japan), May 2012.

W. Hartmann and E. Fosler-Lussier, Investigations into the Incorporation of the Ideal Binary Mask in ASR, *in Proceedings of IEEE ICASSP*, (Prague, Czech Republic), May 2011.

R. Prabhavalkar, P. Jyothi, W. Hartmann, J. Morris and E. Fosler-Lussier, Investigations into the Crandem Approach to Word Recognition, *in Proceedings of NAACL-HLT*, (Los Angeles), pp. 725–728, 2010.

W. Hartmann and E. Fosler-Lussier, Investigating Phonetic Information Reduction and Lexical Confusability, *in Proceedings of Interspeech*, (Brighton, England), pp. 1659–1662, 2009.

R. Fox and W. Hartmann, An Abductive Approach to Hand-Written Character Recognition for Multiple Domains, *in The Proceedings of the 2006 International Conference on Artificial Intelligence* (H. Arabnia, ed.), vol. 2, (Las Vegas), pp. 349–355, CSREA Press, 2006.

R. Fox and W. Hartmann, Hand-Written Character Recognition using Layered Abduction, *in the Proceedings of SCSS* (T. Sobh and K. Elleithy, eds.), Advances in Systems, Computing Sciences and Software Engineering, pp. 141–147, Springer, 2005.

FIELDS OF STUDY

Major Field: Computer Science and Engineering

Studies in Automatic Speech Recognition: Prof. Eric Fosler-Lussier

TABLE OF CONTENTS

Page

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1: INTRODUCTION

Additive noise has long been a problem for ASR [27]. The main issue is the mismatch between the training and testing data; when a system has been trained on clean speech, the noise-corrupted speech can differ to the extent that the trained models no longer match. Other issues, such as high energy noise completely masking portions of the speech signal, can also exist. Approaches to robust ASR tend to focus on either modifying the acoustic models to match the testing condition [23, 16] or modifying the signal [6] or features [50, 102] to match the training data.

One approach known as computational auditory scene analysis (CASA) performs this task through source separation based on the principles of human hearing [97]. The sources are separated in the time-frequency (T-F) domain by assigning each (T-F) unit to a source. This suggests that the ideal binary mask (IBM) is a computational goal for CASA [96]. Each T-F unit is masked or unmasked depending on the ratio of speech energy to noise energy. Directly using the IBM to remove the noise source from the signal has been demonstrated to improve speech intelligibility in humans [100], however, the same success has not been found in ASR without the use of more complicated methods [13, 83].

This thesis investigates two problems concerning the use of binary masks for robust ASR. The first issue is the incorporation of the binary mask in ASR. Previous work reported that the IBM could not be used in ASR without compensating for the masked T-F units [12]; the field of missing data ASR has emerged to provide methods for compensation. We evaluate several proposed missing data methods and compare their results to directly using

the IBM. We demonstrate that if the acoustic features are normalized, directly masking the signal without missing data compensation works at least as well as the comparison techniques. This result was likely missed in earlier studies as they did not use any feature normalization techniques [12, 13, 83].

While the IBM does work well for robust ASR, it requires oracle information about the speech and noise signals; in practice the binary mask must be estimated. Previous work in mask estimation has focused on low level features [44]. As our task is robust ASR, we examine methods for utilizing higher level linguistic information provided by the ASR system. We introduce a new type of masking criterion for an ASR-driven binary mask and propose an estimation method that significantly outperforms baseline techniques using low level features. Due to the formulation of our estimation methods, it also allows for iterative refinement of the estimated masks that is not possible with other methods.

The remainder of the thesis is outlined as follows. Chapter 2 provides a brief overview of a standard ASR system; our focus is on the portions of the system that are important for the techniques discussed in this thesis. Chapter 3 provides a more detailed review of robust ASR methods, focusing on the IBM for speech separation. Common spectro-temporal representations and standard missing data techniques are also discussed. Chapter 4 evaluates methods for incorporating the IBM in ASR and demonstrates the effectiveness of directly applying binary masks without missing data compensation.

The remaining chapters focus on binary mask estimation. An alternative masking criterion for the ASR-driven binary mask is presented in Chapter 5 and details methods for using ASR hypotheses for constraining the mask estimation process. Chapter 6 proposes an method for estimating the ASR-driven binary mask and compares performance against standard mask estimation techniques. Chapter 7 proposes a modification to the estimation

technique to allow for an iterative estimation process. Finally, conclusions and ideas for future work are presented in Chapter 8.

CHAPTER 2: AUTOMATIC SPEECH RECOGNITION

As we will make extensive use of Hidden Markov Model (HMM) based automatic speech recognition (ASR) [81], we describe a standard system in some detail. The goal of an ASR system is to transcribe the words uttered in a given acoustic sequence. For modern statistical ASR, this goal is translated to the probabilistic statement

$$\underset{W}{\operatorname{argmax}}\, P(W|X) \tag{2.1}$$

where W is the word sequence and X is the observation sequence representing the acoustic signal. Since modeling $P(W|X)$ directly would be difficult, we can reformulate the problem by using Bayes' Rule and obtain the statement

$$\underset{W}{\operatorname{argmax}}\, \frac{P(X|W)P(W)}{P(X)}. \tag{2.2}$$

The denominator does not change for a given word sequence, so we can safely ignore it here. Now $P(X|W)$ can be obtained from an acoustic model and $P(W)$ is computed by a language model. In Section 2.1 we describe the standard feature extraction process. Acoustic modeling through an HMM and language models are described in Sections 2.2 and 2.3. Finally, in Section 2.4 we discuss decoding the HMM and the creation of the word lattices we make extensive use of in subsequent chapters. More detailed discussions of large vocabulary speech recognition systems can be found in [104] and more recently in [22].

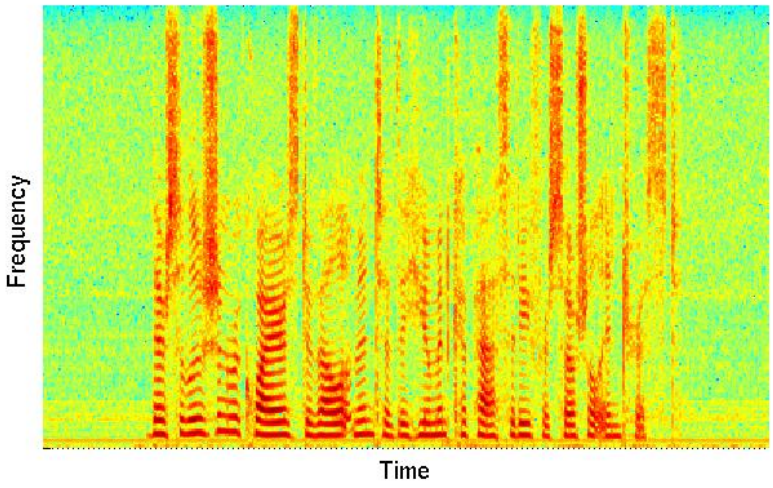


Figure 2.1: An example spectrogram where the x-axis is time and the y-axis is frequency.

2.1 Feature Extraction

In order to better understand how the speech enhancement techniques discussed later can affect ASR, we first give a detailed account of the feature extraction process used in a typical ASR system. We will focus on cepstral features, as they are the most widely used in speech recognition. Specifically, we will discuss mel-frequency cepstral coefficients (MFCC), but the two key points apply to other cepstral features as well. First, our acoustic models assume that the feature input is decorrelated.[1] The cepstral features satisfy this assumption since the final step in their creation is the application of a decorrelation technique. Second, altering a single element in the spectral domain potentially affects all of the cepstral coefficients. The ramifications are discussed in Section 3.4.

[1]Our acoustic models are Hidden Markov Models (HMM) using Gaussian Mixture Models(GMM) with diagonal covariance matrices. This acoustic model is described in more detail in Section 2.2. Other types of acoustic models may not require the features to be decorrelated.

Given a time domain signal, the first step is to transform the signal into the spectral domain, accomplished by applying the Fast Fourier Transform (FFT) to the signal over a short window. Implicitly this assumes that the signal is stationary over that short time window. While this is certainly not true in all instances, this technique works well in practice. Figure 2.1 shows an example of a spectrogram which is a visualization of the amplitudes of the frequencies in the spectral domain. The spectrogram is a matrix where each element is referred to as a time-frequency (T-F) unit. These spectral coefficients can be used as features in a speech recognition task, but they have been shown to not perform as well as cepstral features [15].

The frequency bins in the spectral representation produced by the FFT are evenly spaced. In order to mimic the human auditory system, we apply a filter bank to shift the center frequencies of the bins. While the motivation is based on the human auditory system, previous work has shown a nonlinear filterbank performs better than a linear filterbank [15]. For MFCCs the filter bank represents the mel-scale which is approximately linear to 1000Hz and logarithmic after that; this gives more low frequency resolution and decreases high frequency resolution [45]. Another type of filter bank commonly used is the gammatone filterbank [77, 97]. It also mimics the frequency response of the human ear and cepstral features that make use of the gammatone filterbank have been developed [87]. The spacing of the frequencies in the gammatone filterbank and mel-scale are similar, but the gammatone filterbank becomes logarithmic at a smaller frequency than the mel-scale.

The final step is to apply the discrete cosine transform (DCT) to the log of the spectral features. In addition to transforming the features into the cepstral domain, the DCT decorrelates the features. Cepstral features can be thought of as a representation of the frequency of the frequencies, also known as the quefrency; this term is used to disambiguate

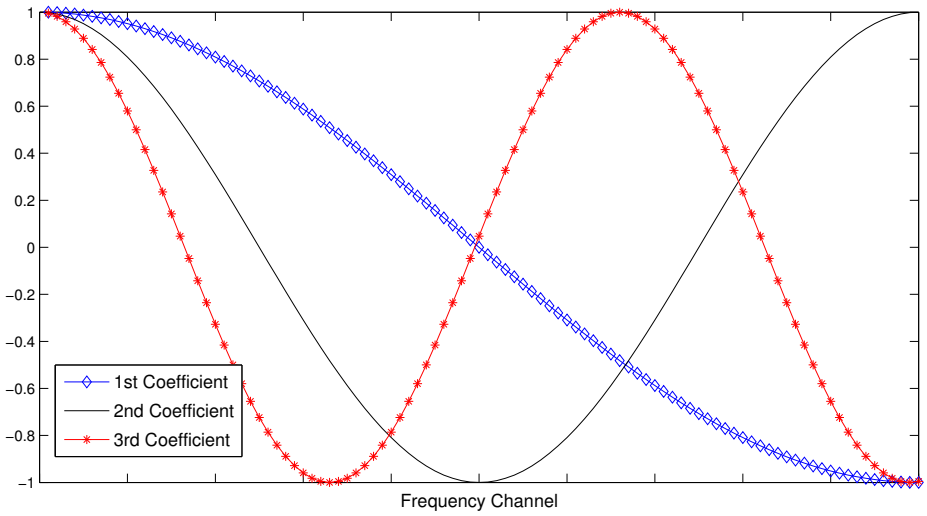


Figure 2.2: Cosines associated with the first three cepstral coefficients. The 0th coefficient would be a line at one representing total energy.

it from the frequency information in a spectral representation. If we consider the source filter model of speech [48], the source corresponds to periodic voiced excitation while the filter corresponds to the vocal and nasal tracts. While the source and filter are convolutive in the time domain, they are additive in the log-spectral domain [38]. The transformation into the cepstral domain allows us to separate a periodic source from the filter [73].

When examining the spectral vector, the periodic source will contribute high quefrency information. This is due to its relatively low frequency contribution in the time domain and its many harmonics. By contrast, the filter will contribute mostly low quefrency information in the spectral domain. Low order cepstral coefficients contain information about the low quefrency portions of the spectral signal. Therefore, by using the low order cepstral coefficients as our features, we are extract features that are mostly influenced by the segmental content of the signal [45].

To better visualize the cepstral features, they can also be thought of as weighted linear combinations of the log spectral features. Weights are determined by a cosine wave with a frequency corresponding to a specific cepstral coefficient. The greater the coefficient index, the higher the frequency of the corresponding cosine. Figure 2.2 shows the cosine corresponding to the first three cepstral coefficients where the 0th coefficient is simply the total energy.

In addition to the standard cepstral features, dynamic features, known as delta and acceleration features, are also typically appended to the feature vector [21]. Delta features are the first derivatives and acceleration are the second derivatives of the cepstral features. Combined, these three feature types constitute the standard feature vector in many ASR systems.

A common final step in feature calculation is mean subtraction [102] and variance normalization. While not all systems perform variance normalization, it can significantly improve recognition [67]. First and second order statistics are calculated over the data and are used to alter the data such that it is zero mean with unit variance.

Given these steps in feature extraction, the interconnectedness of the data is clear. Later we describe speech enhancement techniques that make changes to the spectral features. Even a small change to one T-F unit will affect every cepstral coefficient in that frame. These effects will in turn modify the delta and acceleration features of the surrounding frames. Finally, the changes to these features will affect the calculation of the normalization statistics used to modify every frame. A single change to one spectral value can potentially propagate to every feature in an utterance.

2.2 Acoustic Modelling

As previously mentioned, the acoustic model is used to calculate the $P(X|W)$ term in Equation 2.2. Note that while the term W can represent a single word, it represents a sequence of words when dealing with continuous speech. The acoustic model term can then be represented as

$$P(X|W) = \prod_{i}^{N} P(X_{S_i}|W_i) \tag{2.3}$$

where N is the number of words and S_i is the list of consecutive acoustic features associated with word W_i. This representation is acceptable when you are only concerned with recognizing a limited vocabulary. Consider the case where your vocabulary is several thousands of words; it becomes impractical to represent each word with its own acoustic model not only due to the number of models required, but also because of the amount of training data required. Multiple training examples would be required for each word, implying that it is impossible to handle unseen words.

Instead of modeling words as a singular element, we can model their phonetic representations. For instance, the word *cat* could be represented phonetically as /k ae t/. Each phone would have its own acoustic model that could be shared by multiple words. Now the word *cats* could be represented by /k ae t s/ where the models for the first three phones would be identical to the ones used to represent the word *cat*. We extend our original statement in Equation 2.1 to get

$$\underset{W}{\operatorname{argmax}} \sum_{Q} P(X|Q)P(Q|W)P(W) \tag{2.4}$$

where Q is a phone sequence and $P(Q|W)$ is the pronunciation model. Typically the pronunciation model is handled by a pronunciation dictionary that is simply a mapping from words to their phonetic representations.

In practice, the acoustic model is further refined to model triphones instead of phones. Triphones refer to a representation of not only the phone, but its context as well. Contrasting our previous example of the word *cat*, the new phonetic representation would be /k+ae k-ae+t ae-t/ where /k-ae-t/ refers to the phone /ae/ that has been preceded by /k/ and followed by /t/. The expectation is that the triphone would be a more accurate model since the realization of a phone is dependent on its phonetic context. The use of triphones increases the number of acoustic models that are used.

While many representations for the acoustic model exist, this thesis is concerned only with the Gaussian mixture model (GMM) representation used within the Hidden Markov Model (HMM) framework. Rabiner gives an extensive review of the use of HMMs in ASR in [81]. We briefly discuss the specifics of the GMM as it will be important to our discussion of prior-based spectral reconstruction in Section 3.4.2 and baseline binary mask estimation algorithms in Chapter 6.

The equation for the multivariate Gaussian is

$$N(\mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\right) \tag{2.5}$$

where n is the size of the feature vector, μ is the mean, and Σ is the covariance matrix. In order to reduce the number of parameters we assume that the covariance matrix is diagonal which forces the assumption discussed in Section 2.1 that our features are uncorrelated.

To further improve our representation we use a mixture of Gaussians instead of a single Gaussian. The GMM equation is

$$N(c, \mu, \Sigma) = \sum_{m=1}^{M} c_m N(\mu_m, \Sigma_m) \tag{2.6}$$

where M is the number of mixtures and c_m is the weight for the mth mixture.

The HMM provides a framework for handling sequences of phonetic acoustic models. In fact, the HMM is also used to further refine the phonetic models. By subdividing the model into three states, we can separately model the onset, offset, and steady state portion of a phone. Staying with our previous example, we could talk about the second state of k-ae+t and refer to it as k-ae+t[2]. The HMM is defined by the initial state probabilities, transition probabilities, and observation probabilities. Initial state probabilities are usually defined to either give every state an equal likelihood or defined to have a single default start state. Transition probabilities are typically represented as exponential distributions where the valid transitions at any state are defined by the pronunciation dictionary. Using the previously defined GMM, we can define the observation probability as

$$b_j(X) = N(c, \mu, \Sigma) \tag{2.7}$$

where j is the state index.

The set of probabilities defining the HMM can be learned through an expectation-maximization (EM) algorithm known as the Baum-Welch algorithm [4]. We have now briefly defined the first half of Equation 2.2, the acoustic model. Both the implicit segmentation of the signal that follows from the acoustic model representation and the observation probabilities will be important in later chapters. We discuss the second half of Equation 2.2, the language model, in the following section.

2.3 Language Model

The second half of Equation 2.2 requires a representation for sequences of words and is referred to as the language model. Even in the case of human listeners, context can be important. Early perceptual studies established that ungrammatical sentences are more difficult to recognize when spoken quickly compared to grammatical sentences of identical vocabulary size [66]. In addition, non-linguistic information can affect the speed and accuracy of human spoken language understanding [94]. The language model allows for linguistic context to affect the final output of the recognizer.

Given a word sequence of length K, the probability of the sequence can be represented as the joint probability of the words. This probability can be decomposed into a product of conditional probabilities,

$$P(W) = P(w_1, w_2, ..., w_K) = \prod_{k=1}^{K} P(w_k | w_1, ..., w_{k-1}). \tag{2.8}$$

Unless the possible length of word sequences is severely limited, the number of conditional probabilities required quickly becomes infeasible to model directly. A common assumption used in many language model implementations is the probability of any given word is conditioned only on a small number of previous words. N-gram language models use this assumption where the number of previous words is $N - 1$.

Our experiments later in this thesis mostly use bigram language models; a bigram language model represents the word sequence as

$$P(W) = P(w_1, w_2, ..., w_K) = \prod_{k=1}^{K} P(w_k | w_{k-1}). \tag{2.9}$$

As with the first order markov assumption used by the HMM, this simplification is incorrect. As a simple example, consider the sentences "A deer eats grass." and "A deer eat grass." Only the first sentence in grammatical, but a bigram language model would allow both sentences. According to Google NGrams [65], the ungrammatical sentence would actually be more likely since $P(eat|deer) > P(eats|deer)$ and $P(grass|eat) > P(grass|eats)$. An N-gram of at least order 3 would be needed for this example due to the ambiguity of the count of the noun deer. Similar examples can be contrived for even higher order N-grams.

While examples can be produced where this simplification fails, it is very useful in practice by allowing the incorporation of some context information without requiring an unmanageable number of parameters. The use of the language model is important in ASR and can overcome some of the limitations of the acoustic models, especially in the presence of noise. Table 2.1 demonstrates the effects of varying language models on a large vocabulary task in a noisy environment; results are from the Aurora4 corpus [75] which consists of clean speech mixed with six different types. A more detailed description is presented in Section 4.3.2. As the order of the N-gram model decreases, the accuracy of the ASR system quickly degrades. We make use of the constraining ability of the language model, even in noisy conditions, in later chapters.

The details of training these language models is beyond the scope of this thesis. A more detailed discussion of language models can be seen in [28]. Language models are not limited to generative N-gram models and more recent work has used neural networks [5] and support vector machines [10]. Some work has even incorporated non-linguistic information for speech recognition in video [20].

Language Model	car	babble	restaurant	street	airport	train
Bigram	72.7%	65.7%	63.3%	60.7%	65.0%	58.0%
Unigram	46.1%	41.1%	40.6%	35.7%	39.5%	35.1%
Uniform	19.4%	23.4%	30.7%	28.7%	25.1%	30.6%

Table 2.1: Word accuracy results on Aurora4 using a bigram, unigram, and uniform language model. Each column refers to a separate noise condition mixed with clean speech.

2.4 Decoding

We have briefly outlined the major pieces of an ASR system, the features, acoustic model, and language model. Our goal is to find the word sequence that maximizes Equation 2.2. One possible approach is to evaluate every possible word sequence to find the most likely sequence. However, given a large vocabulary, the number of possible sequences quickly becomes too large to be feasible. Instead, we require an algorithm that can quickly find the best sequence.

Due to the first order markov assumption made by the acoustic model, we can use a dynamic programming algorithm known as the Viterbi algorithm [81]. We note that the algorithm does not find the best word sequence, but the best sequence of HMM states. By taking only the best state sequence for any word sequence instead of summing over all possible state sequences for a word sequence, the computation is simplified and becomes much faster. The difference between the two results is typically seen to be unimportant [38].

To understand how the Viterbi algorithm works, assume that we have a separate HMM for each word in our vocabulary. Each HMM has a sequence of states corresponding to the word's phonetic pronunciation. A matrix is associated with each HMM where each column refers to a frame of speech and each row refers to a particular state. Each value in

the matrix represents the likelihood of the best path to a particular state at a given frame. Decoding becomes the simple process of filling each value in the matrix in a column by column manner.

Using the above algorithm, we can find the most likely sequence given our model. Later in our work it will be important not just to find the best sequence, but a set of highly likely sequences. The set of sequences typically either takes the form of an N-best list or a word lattice. The N-best list, as the name suggests, produces a list of the top N most likely sequences. The word lattice represents this information in a more compact graph structure.

The word lattice is produced in much the same manner as the N-best list. Instead of maintaining a 2-D matrix for each word HMM, we use a 3-D matrix where the third dimension contains multiple likelihoods for a state at a given frame. This allows the algorithm to maintain information about more than just the best sequence at any time. While this increases computational load, it also produces a much richer output that can be used for a variety of applications. In our informal testing, computing a lattice requires about 10x the amount of time as decoding the one best output. However, more recent work has shown the possibility of generating lattices more efficiently [79].

Our brief discussion of decoding glosses over many issues and questions that arise during implementation. More details can be seen in [45] and in the documentation of HMM-based ASR systems such as HTK [105]. We have presented a high-level discussion of HMM-based ASR systems., focusing on the ideas that will be important in the remaining chapters of the thesis. In the next chapter, we provide an overview of robust ASR, where our discussion of both feature calculation and acoustic models will be relevant.

CHAPTER 3: ROBUST ASR

It is well known that the performance of speech recognition systems degrades in the presence of noise. The acoustic models discussed in the previous chapter use statistical models of clean speech. When noise has been added to the clean speech, the statistics of the input data change, creating a mismatch between the observed data and trained models. Many approaches to this problem have been proposed which typically fall into one of three categories: model-based methods, noise-robust features, or speech enhancement [27]. The three categories differ in the stage where the algorithms are applied; model-based methods directly modify the acoustic models, noise-robust features alter the acoustic features, and speech enhancement modifies the signal prior to feature calculation. Our focus is on binary mask based speech enhancement, but we will first outline other methods to illustrate where our approach falls in the range of Robust ASR techniques. After we have described binary mask based speech enhancement, we will discuss how it is used in ASR.

3.1 Methods

The simplest model-based method is to train the system on the noisy speech [30]. While it does provide some improvement over baseline systems, it does not work as well as training and recognizing clean speech. Another drawback is that it typically requires more data depending on the number of potential noise sources and makes some assumptions about the number and type of interfering noise sources.

Current model-based methods attempt to adapt the parameters of the acoustic model to the noisy input speech. Methods such as MLLR [58] and MAP [24] are data-driven and utilize the new input data to adapt the acoustic models. The adaption is not confined to noise robustness, but has also been applied for other domains such as speaker adaptation. Other types of model-based techniques rely not on the data, but on estimates of the noise or channel characteristics. One of the first methods was parallel model combination (PMC) [23]; both clean speech and noise models are trained. By adding the statistics of the noise model to the original clean speech model, an estimated noisy speech model is obtained and used for recognition. The vector taylor series (VTS) approach [69] is another method that accomplishes a similar goal and has recently demonstrated strong performance on a noisy digit recognition task [60].

These model-based methods require some modification to the recognition system or the training paradigm. In addition, they may make some assumptions about the interfering noise. The remaining two approaches modify either the features or original data, removing the necessity of adapting the acoustic models.

The goal of using noise-resistant features is to allow a system that has been trained on clean speech to work equally well on noisy speech without modifying either the acoustic model or the feature vectors themselves. The focus is to mitigate the effects of the presence of noise rather than to remove the noise outright [27]. Perceptually based linear prediction (PLP) features, introduced by Hermansky in [36], provided better performance in noise compared to other features at the time; they were one of the first features considered to have some amount of robustness to noise. RASTA-PLP features produced further improvements by removing logarithmic spectral information that varied slowly with respect to time [37]. More recent features, such as power normalized cepstral coefficients (PNCC) [50], extend

this approach even further. Creating features that are robust to noise is a difficult challenge. Noise-robust features do not affect the trained speech models and allows the use of clean speech training data, but it does force a specific feature type on the acoustic model. Since noise-robust features define a specific feature, the ASR system must be trained on that noise-robust feature.

Speech enhancement algorithms modify either the input signal or feature vectors. Separating speech enhancement algorithms from noise-robust features can be an arbitrary decision; many noise robust features achieve robustness by incorporating enhancement methods in the feature calculation process. The goal of the modification is to bring the statistics of the noisy speech closer to those of the clean speech that was used for training. The acoustic model trained on clean speech is then used with the modified features. Since these algorithms attempt to reduce the noise in the signal in various ways, the hope is that this will allow models trained on clean speech to be used on noisy speech. Many speech enhancement algorithms have been proposed [29, 49, 57], but we will focus on the Ideal Binary Mask (IBM) [40, 62] based speech separation.

3.2 Ideal Binary Mask

The Ideal Binary Mask has been proposed as a goal for speech separation in Computational Auditory Scene Analysis (CASA) [97]. CASA follows the work of Bregman that explained the human ability to analyze a complex auditory scene [7] by segmenting spectro-temporal regions into regions belonging to a common source. A source refers to the originator of a signal, such as a specific speaker. Discontinuous regions that share the same source are then grouped together. This seemingly easy task for humans is difficult

for computers; CASA attempts to apply the same principles used by humans in a computational framework.

Conceptually, the IBM is very simple. A signal is first transformed into a spectrotemporal representation; the spectrogram and cochleagram are two common examples. Each pixel of this two-dimensional image of the signal, or time-frequency (T-F) unit, represents the amount of energy at a particular frequency and time. The IBM is the binary segregation of these pixels into two groups; one containing energy mostly from a target source and one containing energy mostly from the interference. Formally, we can define the IBM as

$$M(f,t) = \begin{cases} 1 & \frac{|S(f,t)|^2}{|N(f,t)|^2} > \theta \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

where f is a frequency band and t represents a particular time frame. $S(f,t)$ and $N(f,t)$ represent the amount of energy at a T-F unit for the clean speech and interfering noise respectively. The threshold θ is typically set to 1 which also corresponds to an SNR of 0 dB. Figure 3.1 shows an example of a noisy signal and its associated IBM.

The IBM segregates the signal where a value of unity indicates that the corresponding T-F unit is grouped into the segregated target, and a value of zero indicates that the unit is considered part of the interference and hence removed [101, 8, 98]. We call T-F units with value 1 *unmasked*, and those with value 0 *masked*. Approaching the problem in this manner reduces speech enhancement to a binary classification task [96]. Perceptual masking studies on humans have shown when the energy from one source is greater than another at a particular frequency band, it effectively masks the source with less energy [68]. Based on those studies, removing the energy in those regions would not remove perceptually relevant information. Previous studies have shown that processing noisy speech using an IBM can

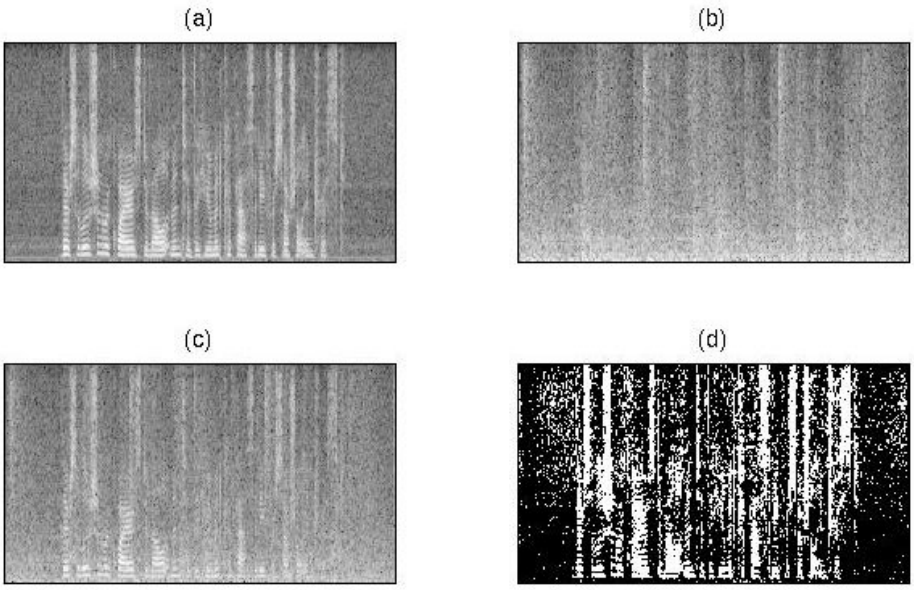


Figure 3.1: (a) Spectrogram of clean speech. (b) Spectrogram of a sample of factory noise. (c) Spectrogram of clean speech mixed with factory noise at 5dB SNR. (d) IBM where the black regions are noise-dominant.

significantly improve speech intelligibility for humans (e.g. [100]) and that the IBM itself carries significant linguistic information [71, 99].

Many methods exist that attempt to calculate the IBM from a mixed signal [41, 44] One general approach follows directly from Equation 5.2. The IBM is estimated by estimating the instantaneous SNR. This can be accomplished by directly estimating the SNR or the individual components of Equation 5.2 separately. A brief overview of this approach can be seen in [43].

The CASA approach to binary mask estimation is another general approach [97]. Estimation is a two step process where small connected groups are segmented. The first stage typically uses perceptually motivated approaches and handles voiced and unvoiced speech

separately [40, 41, 42]. Unvoiced speech is typically viewed as more difficult for speech separation as it does not contain the harmonicity cues found in voiced speech. Once this first stage is completed, segments from the same source are grouped across time and frequency [39].

3.3 Ideal Binary Mask Representations

One important aspect regarding the IBM is the domain in which it is represented. Obviously it must to be a spectro-temporal representation, but many such representations exist. Some representations provide high frequency resolution and high temporal resolution, giving IBMs which inherently carry more information. Others give less resolution, but may be easier to estimate. We briefly discuss the two major representations we consider in this work, the spectrogram and the cochleagram.

3.3.1 Spectrogram

The spectrogram is probably the most common spectral representation for speech. In fact, it is a common representation for any time-domain signal where frequency analysis is required. Given a time-domain discrete signal, the signal is first partitioned into segments. Typically, the segments are overlapping, but it is not required that they be. For each segment, a discrete Fourier transform is performed. This transform can be thought of as a rotation of the space to represent the time-domain segment in a new basis since each dimension is a linear combination of the data in the original time-domain representation. The transformation returns a vector of the same size as the original segment.

The new vector contains values in the complex domain that describe the frequency content of the original segment. Since the spectrogram only contains information about the energy in the frequency bins, only the magnitude of the complex values are used. Every

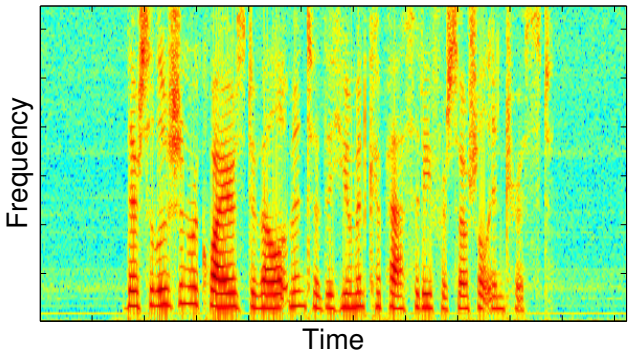
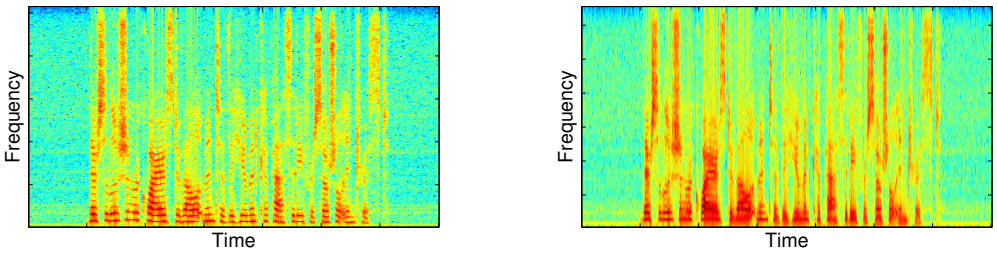


Figure 3.2: An example spectrogram.

value represents the energy in a particular frequency range and every segment corresponds to a time index. An example spectrogram is shown in Figure 3.2.

Each column in the spectrogram corresponds to the log magnitude for the result of the Fourier transform on a segment at a particular time index. The log of the values is typically used to decrease the dynamic range of the spectrogram for visualization purposes. Without the log operation, it becomes difficult to see any detail in the spectrogram. One other important detail is the spectrogram only shows half of each column, due to an effect of the Fourier transform where the values are reflected.

One major question to consider when computing the spectrogram is the size of the window used. A long window will provide fine frequency resolution, but course time resolution. A narrow window will provide course frequency resolution, but fine time resolution. The size of the window provides a trade-off between temporal and frequency resolution. An example of two spectrograms computed with different window sizes is shown in Figure 3.3.



(a) Narrowband Spectrogram (b) Wideband Spectrogram

Figure 3.3: Examples of a (a) Narrowband Spectrogram and (b) Wideband Spectrogram.

3.3.2 Cochleagram

The cochleagram is an alternative spectral representation used widely in CASA [97]. The design of the cochleagram is based on studies of human perception [64, 76, 86]. Instead of performing a Fourier transformation on a windowed signal, gammatone filters [47] are applied to the entire signal, resulting in many time-domain signals. Each signal is the frequency response characterized by a particular gammatone function. To produce the final cochleagram used in our study, each time-domain signal is windowed at regular intervals corresponding to the frames in the representation. The windowed signal at each frame is multiplied with itself to produce the signal energy at that T-F unit.

While the spectrogram produced a representation with a linear frequency axis, the responses from the cochleagram correspond to a nonlinear frequency scale denoted by the ERB scale [26, 77]. Similar to the reasons discussed in Section 2.1 for using the mel-scale for feature calculation, the ERB scale matches properties of the human auditory system. The bandwidth and spacing of the filters is narrow at low frequencies and wide at higher frequencies. An example can be seen in Figure 3.4.

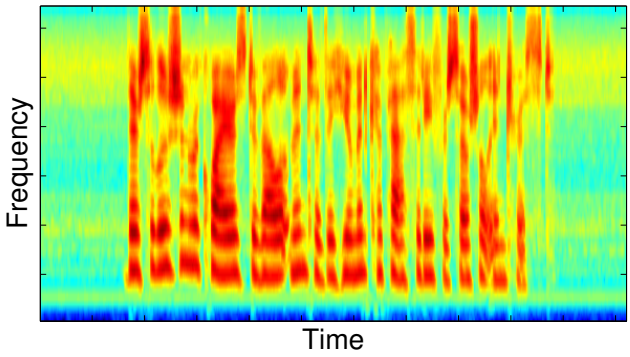


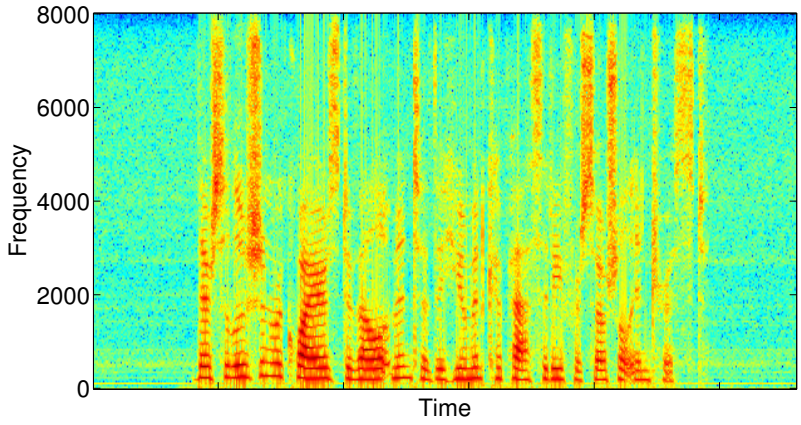
Figure 3.4: An example cochleagram.

Representation	car	babble	rest.	street	airport	train	avg
Spectrogram	13.7%	13.6%	13.8%	14.3%	12.6%	13.8%	13.6%
Cochleagram	17.6%	16.8%	15.2%	18.1%	15.6%	19.1%	17.1%

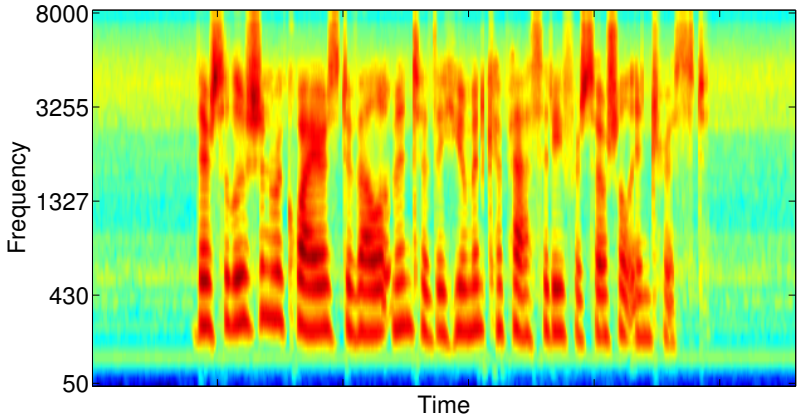
Table 3.1: Word error rates for using the IBM in the spectrogram and cochleagram domain on the Aurora4 dataset. Word Error Rate for clean speech is 9.8%

3.3.3 Comparison of Spectral Representations

For comparison purposes, Figure 3.5 shows the two spectral representations with the frequency axes labeled. Several differences are immediately obvious. The spectrogram contains more granularity in the frequency axis and the formants tend to stand out more. The cochleagram does appear to be smoother and contain less spurious variation. However, the most noticeable difference is in the frequency scale; less than a quarter of the values in the cochleagram correspond to frequencies above 4000 Hz while that frequency range encompasses half the spectrogram.



(a) Spectrogram



(b) Cochleagram

Figure 3.5: Comparison of (a) Spectrogram and (b) Cochleagram representations.

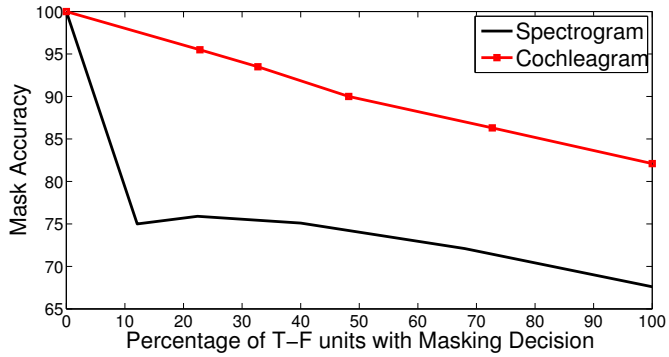
The actual visual differences between the representations are unimportant for our purposes; we are interested in the performance of binary masks in the two domains for ASR. Two properties of these representations are important for this study.

1. The performance of an IBM in a particular domain for robust ASR.

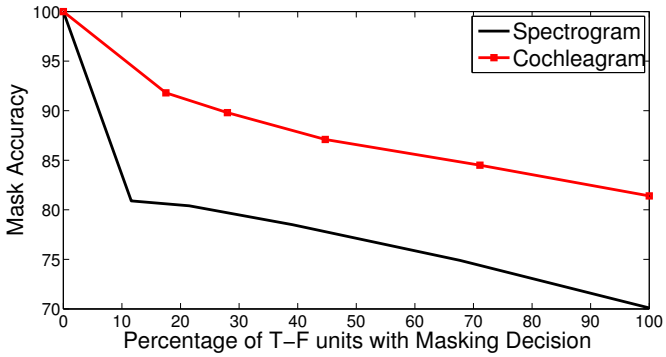
2. The difficulty of estimating the IBM in a particular domain.

While the analysis of these two properties in connection with the spectrogram and cochleagram would take an extensive study, we will briefly present two results that give some insight into the relative strengths of the two representations. Later in this thesis we will describe recognition experiments more fully, but now we just want to present general results. The experimental setup we choose here is very similar to the one that will be described in Chapters 4 and 5. Table 3.1 presents word error rates for several tasks in the Aurora4 dataset when using the IBM with cepstral features. At the moment we are only interested in the comparison between the spectrogram and the cochleagram representations of the IBM; the spectrogram representation performs significantly better than the cochleagram for recognition.

Analyzing the difficulty of estimating the IBM in either the spectrogram or cochleagram domain is a complicated task. Many methods exist to estimate the IBM and it is possible that a new method would completely alter the result of any study. For simplicity, we will only consider the discriminability of a simple spectral subtraction (SS) [6] based method. The SS-based method estimates the SNR at each T-F unit. Since the estimate can vary wildly in accuracy, we will consider the accuracy of the method as a function of the percentage of units estimated; varying the percentage of units estimated is accomplished by manipulating the masking threshold. Results can be seen in Figure 3.6 for clean speech



(a) Factory Noise 10 dB



(b) Factory Noise 5 dB

Figure 3.6: Comparison of mask accuracy results in the spectrogram and cochleagram domain when using a spectral subtraction based mask with various thresholds. Results are on clean speech mixed with factory noise at (a) 10 dB and (b) 5 dB SNR.

mixed with factory noise at 10 dB and 5 dB SNR. As the number of T-F units estimated increases, the accuracy of the estimation decreases, but the drop in accuracy is more stark in the case of the spectrogram. Again, this result cannot definitively state IBM estimation is more difficult in the spectrogram domain, but it does show that simple measures of energy are not as discriminative when compared to the cochleagram.

Based on these results, if a perfect IBM estimator existed, the spectrogram would likely be preferred as it provides better recognition performance. However, if estimation is the task, it may be an easier task in the cochleagram domain. A more extensive study exploring the tradeoffs between mask accuracy and recognition performance would also be needed to make a more definitive conclusion. Our purpose in presenting these results was to simple acknowledge the importance of the spectro-temporal representation when working in the binary masking domain.

3.4 Incorporating the IBM in ASR

Psycho-acoustical studies showing the IBM improves speech intelligibility for humans utilize the IBM in the simplest, most obvious way [9, 100]. Once the IBM has been calculated, the corresponding T-F units are set to zero and the signal is then resynthesized using standard signal processing algorithms [63, 97]. Listeners are presented with speech mixed with noise at low SNRs and are unable to recognize the speech. Once the mixture has been masked by the IBM, the listeners are able to understand the speech. In these perceptual studies, the mask is used directly without any compensation for the T-F units with zero energy. Unfortunately, using the IBM in ASR does not appear to be so simple. We note that datasets used in ASR and human speech intelligibility studies typically differ, so results are not directly comparable. Cooke et al. [12] first examined using the IBM in ASR and found

standard ASR techniques needed to be adapted to deal with the masked T-F units. This has spawned an entire subfield called Missing Data ASR. We describe two of the original methods for Missing Data ASR below.

3.4.1 Marginalization-Based ASR

Originally proposed by Cooke et al., marginalization [13] was the first approach to address the issue of incorporating binary masks in ASR. While several variations were described in [13], we will focus here on the best performing method–bounded marginalization. Features are partitioned into reliable and unreliable ones based on a binary mask. Masked T-F units correspond to unreliable and unmasked units to reliable features. The marginalization-based speech recognizer is a modified HMM-GMM based speech recognizer that treats these masked and unmasked units in separate ways.

In a typical HMM based recognizer, every state is modeled by a GMM. The likelihood of a feature vector X given a particular state Q_i can be obtained by evaluating $p(X|Q_i)$. By separating the feature vector into reliable components X_r and unreliable components X_u, the evaluation becomes

$$p(X|Q_i) = \int p(X_r, X_u|Q_i) dX_u \tag{3.2}$$

where we integrate over (i.e. marginalize) the possible values of X_u. As we are using a GMM for modeling, this becomes

$$p(X|Q_i) = \sum_{c=1}^{M} p(c|Q_i) p(X_r|c, Q_i) \int p(X_u|c, Q_i) dX_u \tag{3.3}$$

where c is a particular Gaussian and M is the number of Gaussians in the GMM.

Just as we partitioned the feature vector into reliable and unreliable portions, we can partition the means and variances of each Gaussian. We can then evaluate $p(X_r|c, Q_i)$ by evaluating the Gaussian only over the reliable dimensions. If we do not assume anything about the unreliable data, then the integral evaluates to one. However, we can at least determine bounds of the true feature based on the unreliable vector. Assuming that X represents speech energy, then the true speech cannot have negative energy or more energy than in X_u. Note that while this assumption can sometimes be violated due to phase interactions, this effect is commonly ignored in missing data literature. because if the speech and noise are independent, then the expected value of phase contribution is 0 [84]. The integral can then be evaluated using these bounds for a more accurate result.

Assuming that a given binary mask is accurate, the marginalization-based recognizer utilizes the available information from all the T-F units. Reliable units are treated in the standard way and unreliable features provide bounds on marginalization. On small vocabulary tasks such as TIDigits [59], the marginalization approach performs remarkably well. However, performance on larger vocabulary systems degrades significantly [83, 89]. A likely cause is the use of spectral features instead of the cepstral features which are known to perform better in ASR[15]. Methods that allowed for the calculation of cepstral features were needed to further increase performance, at least for larger vocabularies.

3.4.2 Reconstruction-Based ASR

One method that allows for the calculation of cepstral features is the estimation or reconstruction of missing T-F units. If the missing T-F units can be reconstructed, then the zeros or holes in the spectral representation no longer present a problem for cepstral feature

calculation. The first comprehensive study of feature reconstruction was presented by Raj et al. [83].

It was clearly shown that this method only provided improvements over marginalization when using cepstral features. If instead the recognition was performed in the spectral domain, the reconstructed features performed worse. The results also held over larger vocabulary tasks. Another benefit of this technique is that it does not require any modification to a standard recognizer.

Many specific techniques for performing the reconstruction have been explored [83, 25, 85]. We will present a technique that has been previously shown to improve results over a baseline system [91]. As with marginalization, a binary mask is used to partition the noisy speech vector Y into a reliable set Y_r and an unreliable set Y_u where $Y = Y_r \cup Y_u$ and $Y_r \cap Y_u = \emptyset$. Given Y, we want to estimate the true spectral vector $\hat{X}$ for the clean speech.

Assume $X_r = Y_r$. In order to estimate X_u, a speech prior is used [83]. The speech prior, consisting of spectral features instead of the cepstral features eventually used for recognition, is modeled by a GMM. Just as we used the binary mask to partition the spectral vector, we can also use it to partition the mean and covariance of each mixture.

$$\mu_c = \begin{bmatrix} \mu_{r,c} \\ \mu_{u,c} \end{bmatrix} \quad \Sigma_c = \begin{bmatrix} \Sigma_{rr,c} & \Sigma_{ru,c} \\ \Sigma_{ur,c} & \Sigma_{uu,c} \end{bmatrix} \tag{3.4}$$

Ideally we would select the Gaussian that generated the spectral vector for estimation. Since we cannot identify the specific Gaussian, the estimate is the weighted sum of the estimates from each Gaussian.

$$\hat{X}_u = \sum_{c=1}^{M} p(c|X_r)\hat{X}_{u,c} \tag{3.5}$$

where M is the number of Gaussians and $\hat{X}_{u,c}$ is the expected value of X given the cth Gaussian. To estimate $p(c|X_r)$, the marginal distribution $p(X_r|c) = N(X_r; \mu_{r,c}, \Sigma_{rr,c})$ is used [91]. Finally, we compute the expected value of X_u given the cth Gaussian by

$$\hat{X}_{u,c} = \mu_{u,c} + \Sigma_{ur,c}\Sigma_{rr,c}^{-1}(X_r - \mu_{r,c}). \tag{3.6}$$

The unreliable portion of the spectral vector is then replaced by the estimate $\hat{X}_u$ and cepstral features are computed from the reconstructed spectrogram.

Given that this approach allows for the calculation of cepstral features, performance is expected to scale to any size vocabulary. As methods for improving reconstruction further develop (e.g. [53]), ASR results should also improve.

CHAPTER 4: RE-EVALUATING MISSING DATA ASR

In the previous chapter, we discussed the incorporation of the IBM in ASR. The original study by Cooke et al. [12] demonstrated that directly using the IBM in ASR did not work; thus, missing data ASR was born. This result went largely untested and unexamined over the next decade and a half. We first examined the direct use of the IBM in [31]. In this chapter we show that the direct use of the IBM can be efective for robust ASR. Previous work may have missed this result due to the lack of variance normalization on the cepstral features. We also compare the direct approach to two previously proposed missing data ASR methods.[2]

4.1 Introduction

While Cooke et al. [12] showed standard ASR techniques needed to be adapted to handle IBM enhanced speech on the TIMIT task, no investigation or explanation regarding this phenomenon was given. In a later study, they test the direct masking approach on the Resource Management task [80], but provided no explanation for its poor performance [14]. Since their study, the field of missing data ASR has greatly advanced. However, no further investigation has been made regarding the direct use of binary masked speech and no work in missing data ASR offers it as a baseline.

[2]Much of this work was done in conjunction with Arun Narayanan, Eric Fosler-Lussier, and DeLiang Wang and has been submitted to the *IEEE Transactions on Audio, Speech, and Language Processing* journal [33].

Most studies simply assume directly using the masked speech is not feasible and do not even present an explanation for the necessity of missing data ASR. If an explanation is given, it is usually stated that the cepstral transformation smears the uncertainty introduced by the mask over every cepstral feature [91]. While this is true and we did discuss this issue in Section 2.1, the same thing could be said for any technique that operates in the spectral domain. Any change made to a T-F unit will propagate to every cepstral feature in that frame. The question remains why masking T-F units affects the features more strongly than other methods.

In the remainder of this chapter we will show the IBM can be used directly without the use of missing data ASR. We also compare the results with the two missing data ASR techniques described in Section 3.4. Our results show directly using the mask performs favorably compared to the two missing data techniques on both a small and large vocabulary task. Section 4.2 describes directly using the IBM, which we term the direct masking approach. Comparisons between the direct masking approach and the two missing data methods on the TIDigits [59] task and the Aurora4 [75] task are shown in Section 4.3. Analysis explaining the differences between our results and previous work is presented in Section 4.4 and concluding remarks are presented in Section 4.5.

4.2 Direct Masking Approach

Most methods for incorporating a binary mask in ASR begin from the same place. Given a binary mask, estimated or ideal, the signal is masked in the spectral domain. At this point the various methods diverge as they attempt to compensate for the masked regions in various ways. The marginalization method described in Section 3.4.1 modifies the recognizer to marginalize over the masked spectral values. The reconstruction-based

method described in Section 3.4.2 attempts to estimate the missing spectral values based on the unmasked T-F units. Instead of compensating for the masked T-F units, the direct masking approach simply treats the missing values as zero and continues the feature calculation process as usual. Other work has referred to this approach as zero-imputation [14], but we feel the term direct masking better captures the distinction between this approach and other missing data approaches as no actual imputation or marginalization takes place.

Stated explicitly, the approach takes a binary mask corrupted in the spectral domain and multiplies the mask by the representation of the signal in the same domain. We note that instead of leaving the masked regions as zero, a noise floor could also be used. From the masked spectral representation of speech, we can either directly calculate features or resynthesize the waveform signal. In our study, we resynthesize the waveform prior to calculating features. First resynthesizing the waveform is useful as it allows for more flexibility in the parameters used in computing the binary mask. Resynthesis is a standard procedure and can be accomplished through the overlap-add method [63] for spectrograms or the methods described in [97] for the cochleagram.

Once again, the difference between this approach and the previously described approaches is that no attempt is made to modify the signal, features, or recognition system to compensate for the artificial zeros introduced in the spectral domain. Calculation of features from a signal with artificial zeros could possibly be a problem; most standard ASR features require a log operation and the log of zero is undefined. However, standard MFCC and PLP calculation software [105, 19] typically add a small amount of dither or noise to the signal prior to the log operation being performed. Even if the feature calculation does not perform this step, the masked regions could be set to a noise floor instead of using an explicit zero.

Results	Description
Figure 4.1	Compares results using the cepstral (PLP), linear spectral (spectrogram), and nonlinear spectral (CRate64_D) on TIDigits.
Figure 4.2	Marginalization, reconstruction, and direct masking results on TIDigits task where the IBM has been calculated in the linear spectral domain.
Figure 4.3	Marginalization, reconstruction, and direct masking results on TIDigits task where the IBM has been calculated in the nonlinear spectral domain.
Table 4.3	Comparison of reconstruction and direct masking on the Aurora4 corpus.
Figure 4.4	Average results on the Aurora4 corpus for direct masking and reconstruction with randomly perturbed masks.
Figure 4.5	Results comparing direct masking and reconstruction with randomly perturbed masks on each of the six noise types in Aurora4.
Figure 4.6	Demonstration that variance normalization significantly improves results when using the direct masking approach.
Figure 4.7	Comparison of the variance of cepstral features for clean speech, noisy speech, and IBM masked speech.

Table 4.1: Outline of experiments presented in Section 4.3.

Outside of Cooke et al [14], we have been unable to find a study using this simple approach. The likely cause is that results were sufficiently poor in [14] to convince subsequent researchers the direct masking approach does not work. Since the study, the conventional wisdom has been that this direct masking approach does not work. Given this assumption, much investigation has been made into alternative methods of utilizing binary masks in ASR. We examine the validity of this assumption in the following sections.

4.3 Re-evaluation Experiments

Our experiments parallel the early work in missing data ASR. We compare the results of bounded marginalization and spectral reconstruction on both a small and large vocabulary dataset to the direct masking approach. The original study using marginalization by Cooke et al. [13] reported results on TIDigits [59]. We will attempt to reproduce their results and compare them to the direct masking approach. For the larger datasets, we will use the Aurora4 corpus [75]. The Aurora4 corpus is a 5000-word closed vocabulary task, generated by adding noise to clean speech recordings in the Wall Street Journal (WSJ0) database [78]. Each utterance has been mixed with a noise source at a randomly chosen SNR between 5 and 15dB. In total, six different noise types are used. As spectral reconstruction was only shown to outperform marginalization on larger datasets [83], our focus will be on reconstruction for the Aurora4 dataset. Our recognition systems are similar to the ML-trained systems used in [13, 83, 91]. We acknowledge our baseline results are not state of the art, but our goal was to have a fair comparison to the previously discussed studies. Since we are presenting a large number of results, we provide Table 4.1 as an experimental map, briefly describing each experiment and providing a link to the results.

4.3.1 TIDigit Experiments

The TIDigits corpus is a speaker independent connected digit recognition task. The training set consists of 8623 utterances spoken by 55 male and 57 female speakers. The test set is of similar construction and consists of 8700 utterances spoken by a separate set of 56 male and 57 female speakers.

For the marginalization-based experiments, we attempted to match the study of Cooke et al. [13] as closely as possible. Where differences exist, we will make an explicit note.

The first difference is we use the entire test set as opposed to the 240 utterance subset used in [13]. In addition to recreating the marginalization-based results, we will also compare the direct masking approach to the reconstruction-based missing data approach.

Since the marginalization-based missing data recognizer cannot use standard cepstral features, we must select a spectral feature to use. In [33], we compared a set of spectral features to identify the best performing feature for the marginalization-based experiments. Two features were selected, one for use with an IBM computed in a linear spectral domain and another for use in a nonlinear spectral domain.

The spectrogram representation of speech has a linear frequency axis and is computed by transforming the time-domain signal to the spectral domain by the FFT. For our experiments, we use a 10ms step size and a 20ms window size with a Hamming window. The spectrogram features are then computed by taking the log of the magnitudes at each T-F unit, giving a 160-dimensional feature vector. Similar spectrogram-based features have been used in [89]. We did not incorporate delta components as they did not provide any performance benefit.

Cubic compressed Rate64 with delta (CRate64_D) features were used when the IBM was computed in a nonlinear spectral domain. First the signal is passed through a 64-channel gammatone filterbank. Next, the Hilbert envelope is extracted at each channel and down sampled to 100Hz. Each T-F unit is then compressed with a cubic root. Similar features were used in [13, 92]. Finally, we add delta components as they have been previously shown to improve missing data ASR performance [3, 88]. While these features differ somewhat from those used in [13], they provide the best performance of all features tested. These features were extracted with the help of the CASA Toolkit [2].

In addition to the spectral features used for marginalization-based experiments, we use cepstral features for the direct masking and reconstruction-based approaches. We use mean-subtracted and variance normalized PLP features with delta and acceleration features comprising a 39-dimensional feature vector. For baseline performance, features are calculated directly from the given test signal. In the case of the IBM experiments, the signal is first segregated using the binary mask. For the direct masking approach, the waveform is resynthesized from the segregated target, while the reconstruction approach first estimates the energy in the masked T-F units. The missing information is estimated from a 1024-component GMM speech prior. Features are then computed from the resynthesized waveform.

The IBM is defined using a local SNR criterion of 0 dB in all three approaches. In [13], they used the *a priori* mask which corresponds to an IBM with a local criterion of 7.7 dB. After experimenting with both criterions, we found the 0 dB criterion produced the best performance. The masks appear similar for all feature types used, but the delta mask for CRate64_D is defined slightly differently. Only when every T-F unit used to compute a delta feature is unmasked is the corresponding delta feature considered unmasked [3].

Our recognizer is composed of 11 word-level models (1-9, 'oh', and zero) and a silence model. The silence HMM is three states while each word-level HMM is 8 states, each consisting of a 10-component GMM. All models are trained on clean speech using HTK [105]. The standard HTK viterbi decoder was modified to also perform bounded marginalization.

We report results on the TIDigits connected digit task. Baseline results for the three feature types can been seen in Table 4.2; as expected the PLP features perform the best. Since the focus is on robust ASR, we create noisy speech data by artificially mixing the clean speech data with noise samples from the Noisex92 database [95]. Three noise types

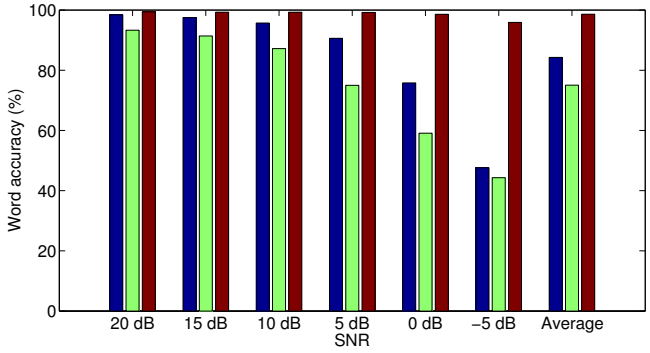
Feature	Feature Domain	Word Accuracy
CRate64_D	Spectral (non-linear frequency axis)	98.7
Spectrogram	Spectral (linear frequency axis)	94.2
PLP	Cepstral	99.2

Table 4.2: Word accuracies obtained using the clean test set of the TIDigits corpus for various features.

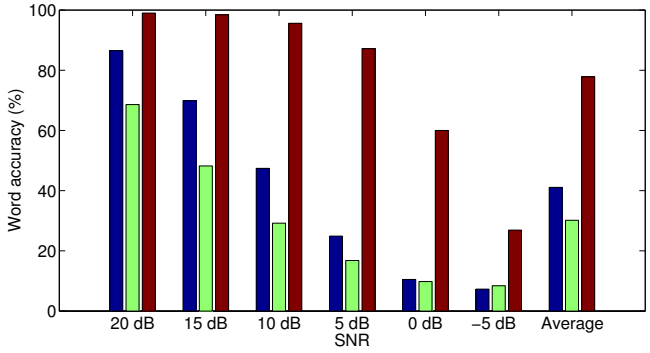
were selected, car, babble, and factory, and were mixed at an SNR ranging from -5 dB to 20dB with a 5 dB step increment. The three noise types and six SNRs gave a total of 18 testing conditions.

Baseline results for the three feature types on all 18 testing conditions can be seen in Figure 4.1. Once again PLP features outperform the other two feature types. In fact, performance of the PLP features at 5 dB is comparable to the performance of the spectral features at 20 dB SNR. We note the performance degradation caused by car noise is far less than that of the other two noise types. The car noise used is relatively stationary and concentrated in low frequency regions.

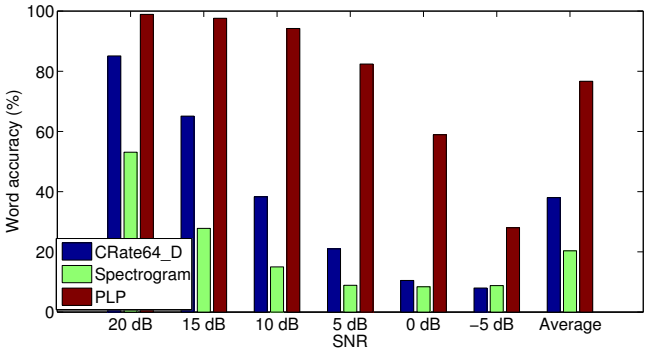
Now that we have established relevant baselines for each of the test conditions, we can explore the results when using the IBM. We first examine the IBM computed in the linear frequency domain, requiring the use of the spectrogram features. Figure 4.2 shows results for all three approaches for incorporating the IBM in ASR. Performance for direct masking and reconstruction are very similar, while the marginalization is much worse on every test condition. Next, we present results using the IBM computed in a non-linear frequency domain in Figure 4.3. Recall this requires the use of the CRate64_D features for marginalization. Performance for all three approaches is similar, but the direct masking approach seems to degrade more quickly than the other approaches at low SNRs. Marginalization



(a) Car noise

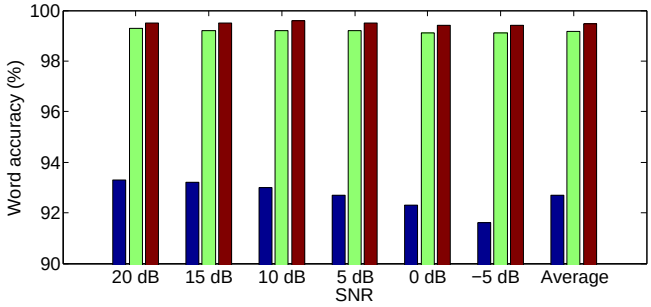


(b) Babble noise

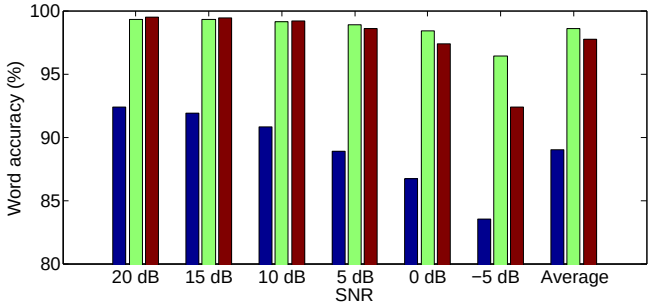


(c) Factory noise

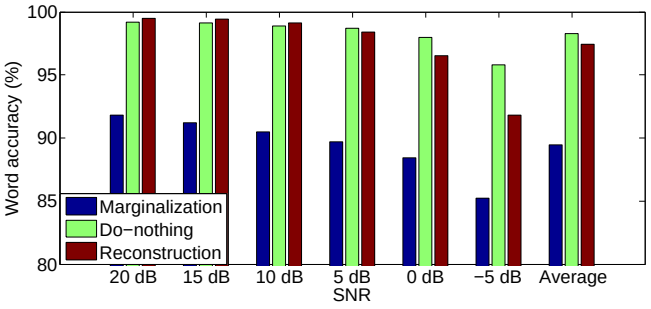
Figure 4.1: Word accuracies in noisy conditions for 3 features and 6 SNR conditions from 20 dB to -5dB, in decrements of 5 dB on TIDigits. Also shown is the average word accuracy for each feature, across all SNR conditions. (a) Car noise. (b) Babble noise. (c) Factory noise.



(a) Car noise

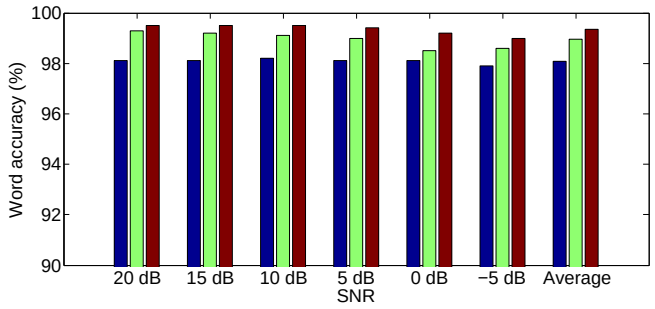


(b) Babble noise

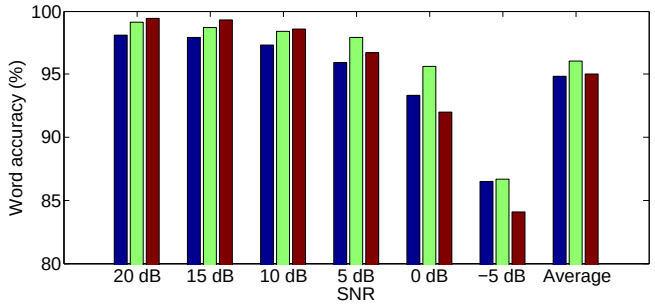


(c) Factory noise

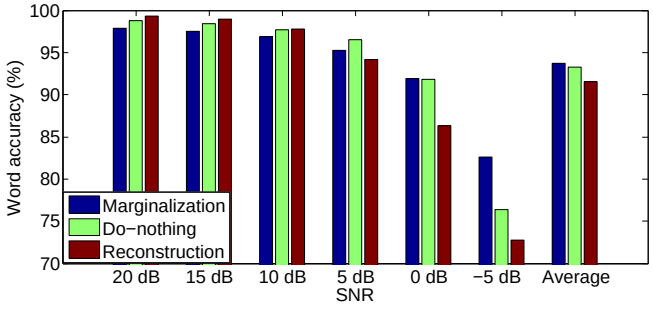
Figure 4.2: Comparison of marginalization, direct masking, and reconstruction in the linear frequency domain on TIDigits. Marginalization uses CRate64_D spectral features. The other two approaches use PLP cepstral features. Also shown is the average word accuracy across all SNR conditions. (a) Car noise. (b) Babble noise. (c) Factory noise. Note the scale of the ordinate.



(a) Car noise



(b) Babble noise



(c) Factory noise

Figure 4.3: Comparison of marginalization, direct masking, and reconstruction in the non-linear frequency domain on TIDigits. Also shown is the average word accuracy across all SNR conditions. Note that the scale on the ordinate does not start at 0.

also performs better in the nonlinear frequency domain. This is likely due to the fact the CRate64_D features are better performing features in general than the spectrogram features. Additionally, this could be seen as a drawback to the marginalization approach as the features are so strongly attached to the way the IBM is computed.

Before continuing, we would like to highlight one key point drawn from these results. At no point does the direct masking approach appear not to be a viable alternative to the other approaches. The results do not provide strong evidence that reconstruction or marginalization are significantly better than the direct masking approach. We examine whether these conclusions hold on a larger dataset in the next section.

4.3.2 AURORA4

Our experimental setup here is very similar to the one used in the previous set of experiments. The Aurora4 [75] corpus is a 5000-word closed vocabulary task. It was generated by adding noise to clean speech recordings in the Wall Street Journal (WSJ0) database [78]. Each utterance has been mixed with a noise source at a randomly chosen SNR between 5 and 15dB. In total, six different noise types are used. Note that Aurora4 does not allow for a breakdown by SNR as each test set contains a mix of SNR conditions.

Using the HTK toolkit [105], we trained a baseline HMM recognizer on clean speech. Our models consisted of tied-state inter-word triphones with 16 Gaussians per state. The CMU dictionary was used for our pronunciations. Cepstral mean and variance normalized PLP features with delta and acceleration coefficients were used, giving a 39-dimensional feature vector. The reconstruction speech prior, consisting of a mixture of 1024 Gaussians, was also trained using the HTK. Again, the IBM was generated by comparing the premixed

System	car	babble	rest.	street	airport	train	avg
Baseline	72.7%	65.7%	63.3%	60.7%	65.0%	58.0%	64.2%
Reconstruction	84.3%	83.5%	84.1%	82.7%	84.5%	81.9%	83.5%
Direct Masking	86.3%	86.4%	86.2%	85.7%	87.4%	86.2%	86.4%
Perfect Reconstruction	90.2%	90.3%	90.2%	90.4%	90.7%	90.2%	90.3%

Table 4.3: Word accuracy results using the IBM on the Aurora4 test set. Baseline is the unsegregated noisy speech.

clean speech energy to the noise energy in the linear frequency domain using a local SNR threshold of 0dB.

We performed recognition experiments to compare the use of masked and reconstructed speech.[3] Our results utilizing the IBM can be seen in Table 4.3. Baseline refers to the recognition of unsegregated noisy speech. As expected, the addition of noise causes a significant drop in performance compared to word accuracy when recognizing clean speech, which is 91.7%. Reconstruction refers to speech where the masked regions have been estimated utilizing the technique described in Section 3.4.2. When comparing these results to the baseline, we see a significant improvement. This is the type of comparison typically shown in the literature discussing spectral reconstruction [25, 53, 91]. With such improvements in accuracy over the baseline, it is easy to see how claims about the utility of reconstruction can be made.

However, these two results alone do not tell the whole story. Consider the direct masking results where no attempt to reconstruct masked units has been made. Its performance is better than reconstructed speech in every case. By attempting to reconstruct the missing spectral energy, performance was actually hindered. Combined with the results presented

[3]We did not perform marginanlization-based experiments since the best performing spectral feature performed worse on clean speech than the PLP baseline for any noise. The clean speech results using spectral features sets a ceiling for performance for the marginalization-based recognizer.

on TIDigits, this highlights a major issue within the missing-feature ASR literature. Without a comparison against the direct masking approach, it is unclear whether a particular reconstruction technique provides any benefit.

While our results show the direct masking approach significantly outperforms this particular reconstruction technique, we do not claim that the idea of reconstruction itself is ineffective. More sophisticated techniques could potentially surpass the simple direct masking approach. In Table 4.3 we also show results for perfect reconstruction, where every missing T-F unit has been replaced by the true energy of the clean speech. If the reconstruction worked perfectly, it would significantly outperform the direct masking approach.

We recognize that results using ideal masks may not tell the whole story as estimated masks do contain errors. We now examine the effects of mask errors on our results. Instead of using a specific mask estimation algorithm, we examine the effects of randomly perturbing the ideal mask. Obviously the errors introduced in this manner would differ from the errors seen in a mask estimation algorithm, but it does provide a general idea of the effect of mask errors and has been used in previous studies [14, 91].

Results on the Aurora4 test set are shown in Figure 4.4. Average word accuracy across all 6 noise types for both the direct masking and reconstruction approaches versus energy deviation are shown. Results for each individual noise type can be seen in Figure 4.5. Energy deviation is the ratio of energy in the incorrectly labeled T-F units with respect to the total target energy. We use this metric as opposed to a simple count of unit labeling errors because we expect errors in high energy units to affect the final result more than those in low energy regions; the greater the energy in a T-F unit, the greater its contribution to the resulting cepstral features.

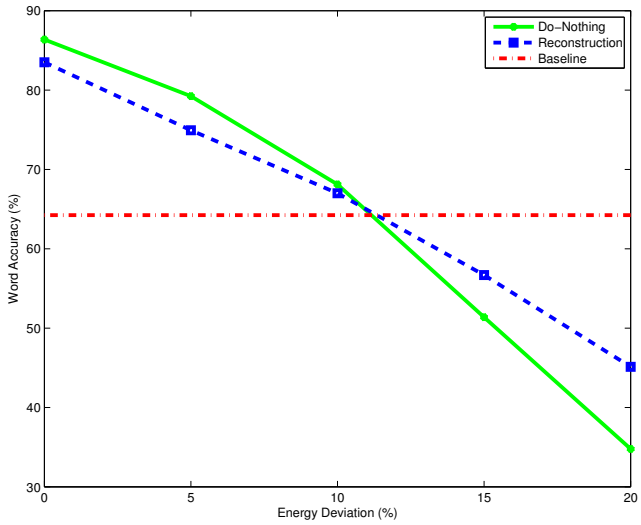
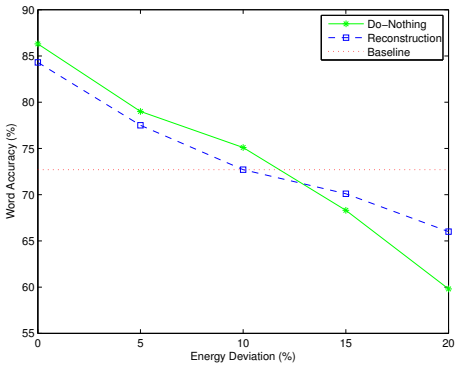


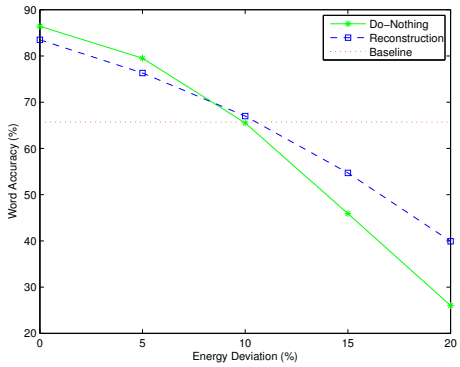
Figure 4.4: Word accuracy results on the Aurora4 test set using randomly perturbed ideal binary masks. Average results over all 6 test conditions are displayed. Results on all test conditions follow the same general pattern. The results use an IBM where a given percentage of energy has been incorrectly classified as speech or noise dominant.

Our results show reconstruction begins to outperform the direct masking approach at around 10% energy deviation on average. However, by the time reconstruction becomes the better performing metric, performance is similar to the baseline. Results on both ideal and perturbed masks provide evidence that the simple direct masking approach does not perform significantly worse than reconstruction. In fact, just a small reduction in energy deviation would produce a larger increase in performance than perfect reconstruction would produce over the direct masking approach. Improvements to mask estimation may provide greater performance improvements compared to improvements to reconstruction methods.

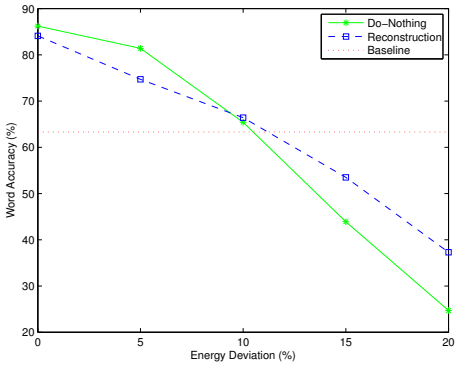
Regardless, future studies should utilize the direct masking approach as a baseline rather than unsegregated noisy speech. We also believe this demonstrates that future work in mask estimation can evaluate performance in ASR without requiring more complicated



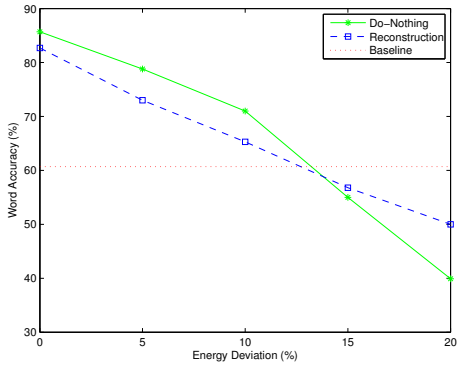
(a) Car noise



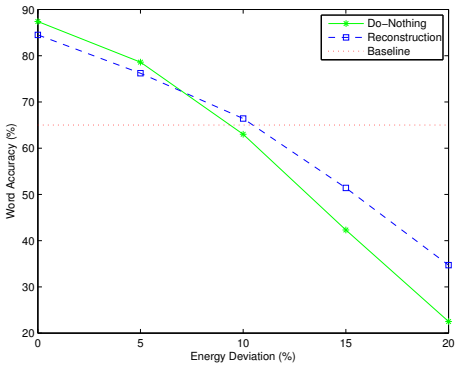
(b) Babble noise



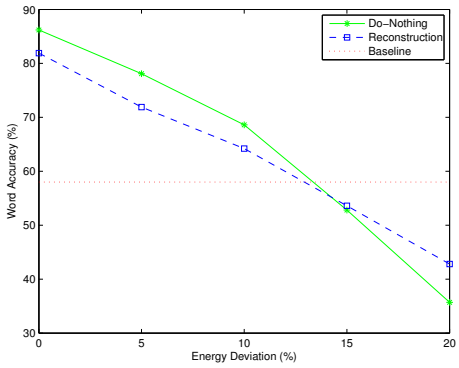
(c) Restaurant noise



(d) Street noise



(e) Airport noise



(f) Train noise

Figure 4.5: Word accuracy results on the Aurora4 test set using randomly perturbed ideal binary masks. The results use an IBM where a given percentage of energy has been incorrectly classified as speech or noise dominant. (a) Car noise. (b) Babble noise. (c) Restaurant noise. (d) Street noise. (e) Airport noise. (f) Train noise.

reconstruction techniques. In the next section we will explain why our experiments showed, in contrast to previously held beliefs, that directly using binary-masked speech can work well in ASR.

4.4 Why is Direct Masking Ignored?

We have established that directly using the IBM can perform well, but why has this been missed by previous researchers? The direct masking approach has been previously tested [14] and results were poor. Other studies likely ignored this approach based on these early results. If this is true, then what is different between our experimental setup and the likely setup of previous work? In our previous study [31], we found correlation between language model strength and recognition performance, suggesting that the Aurora results may have been due to the influence of the language model. However, the present study shows a similar effect for small vocabulary and large vocabulary tasks, indicating that the language model may not be a primary reason.

The remaining difference is the features used. Due to its popularity, previous work likely used MFCCs generated using the HTK. As already mentioned, our experiments used PLP features generated using the ICSI tool Feacalc [19]. In order to test our hypothesis that the feature type could drastically affect the results, we attempted to use the direct masking approach with MFCC features. Results on the TIDigits data mixed with factory noise are shown in Figure 4.6. Performance for other noise types was similar. The direct masking approach using the IBM clearly does not work. In fact, it performs worse than no segregation at all. Obviously if previous researchers had seen a similar result, it would have served as a strong motivator to explore techniques for incorporating a binary mask in ASR.

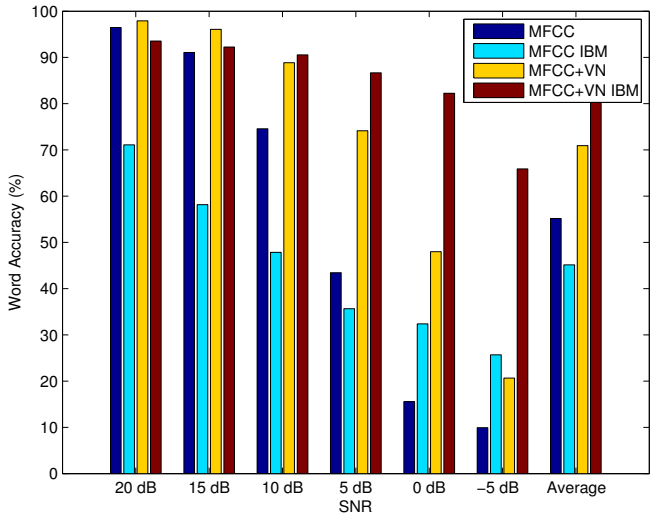


Figure 4.6: Word accuracies for the factory noise condition on TIDigits for MFCCs with and without variance normalization and with and without the IBM. Results are shown for 6 SNR conditions from 20 dB to -5dB, in decrements of 5 dB. Also shown is the average word accuracy for each feature, across all SNR conditions.

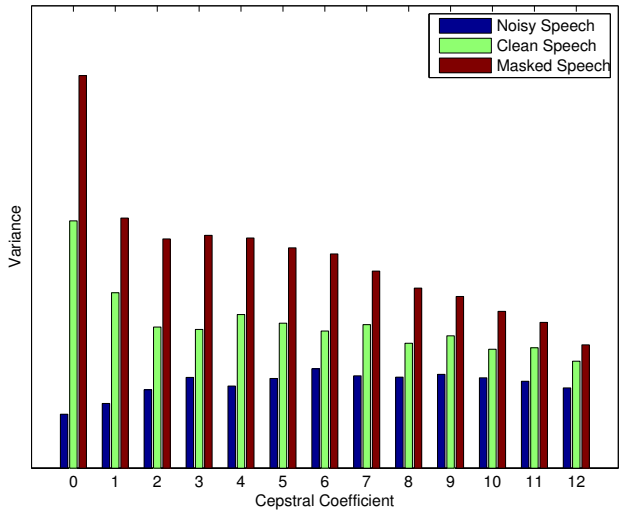


Figure 4.7: Variances for the first 13 MFCCs computed from speech mixed with babble noise, clean speech, and IBM masked speech. The noise was mixed with speech from TIDigits at an SNR of 5dB.

Many differences exist between the two feature types, but we found variance normalization was the only crucial difference; variance normalization sets the variance of each feature dimension to 1 by dividing by the standard deviation. Although HTK-based features typically do not use variance normalization, it is a commonly used technique in the field. To show the effects of variance normalization, we perform it on the MFCC features and show the results in Figure 4.6. Each dimension was normalized to have a unit variance per utterance. Two things are immediately obvious when comparing the results in Figure 4.6. First, variance normalization has improved every result. Even recognition on the noisy speech directly is significantly improved at lower SNRs. Second, the direct masking approach now performs remarkably well. The benefits of using the IBM are only significant at lower SNRs. The variance normalized cepstral features themselves appear to be fairly robust in this small vocabulary domain. Regardless, simple variance normalization allows the direct use of the IBM to be a strong alternative to other techniques.

We can examine the average variance of the first 13 cepstral coefficients of the MFCC features used. Results are shown for babble noise at 5dB SNR for the TIDigits data set in Figure 4.7. The pattern is consistent across all noise types and SNRs. As expected, variances for features generated from noisy speech are less than features generated from clean speech. The noise effectively fills in T-F units with low energy and decreases the dynamic range of the clean speech.

As an illustrative example, consider two phones, /f/ and /ae/, with very different spectral characteristics. The first cepstral coefficient basically captures the difference in energy between the low and high frequency channels. For a frame of /f/, the first cepstral coefficient would be highly negative as most of the energy would be in the high frequency channels. Conversely, a feature calculated from a frame of /ae/ would be more positive due

to the concentration of energy in the low frequencies. By adding noise to the signal, the contrast between the high and low frequencies for /f/ and /ae/ would be reduced, bringing the features closer together. Similar arguments can be made for the other coefficients.

An opposite effect is seen for the features generated from the IBM-masked speech. The variances are consistently greater than the clean speech variances. If noise fills in low energy errors, then masking instead removes energy from these original low-energy regions. The range of the features is expanded by the contrast between the artificial zeros and the areas containing large amounts of speech energy that also keep the additional energy from noise. Considering the same example of /f/ and /ae/, the difference between the high and low frequencies is increased by the artificial zeros, driving the features further apart. In some ways the masking overcompensates for the addition of noise by removing too much energy and the variance normalization corrects for this effect.

4.5 Conclusion

We have shown the commonly held belief that a binary mask cannot be used directly in ASR is incorrect. In fact, directly using the IBM outperforms more complicated techniques on a variety of datasets. Previous work likely missed this result due to the lack of variance normalization on acoustic features. By controlling the variance of the features, even results on the unsegregated noisy speech improved. Since the increase in variance appears to be a major issue, similar ASR systems should include variance normalization. It is possible that some speech enhancement methods simply mitigate this issue.

While much research has been done in missing feature ASR, it may be built on the incorrect belief that the IBM cannot be used directly. We believe the initial work in marginalization and reconstruction strongly influenced the focus of subsequent work. With the

success of missing feature ASR compared to direct recognition of noisy speech and the drive to improve existing techniques, it may be difficult to see a re-evaluation like the one presented here. We certainly do not claim the direct masking approach should replace all missing data methods, however, we believe it presents a much stronger baseline that should be used as a comparison against future methods. Also, future work in speech enhancement may be able to evaluate their methods in terms of ASR performance without needing to implement more complicated missing feature methods.

Based on these results, the focus of the remainder of this thesis will be on binary mask estimation. We will take advantage of the fact that missing data methods are not strictly necessary. Also, we believe work in estimation has the potential for much greater contributions to robust ASR than improved missing data methods.

CHAPTER 5: AN ASR-DRIVEN TOP-DOWN BINARY MASK PROPOSAL

Typical approaches to binary mask estimation use low-level features and bottom-up techniques. One reason that estimation techniques focus on the low-level features is that the IBM itself is defined based on the local SNR at the T-F unit level. We propose an alternative masking criterion that is defined by higher level linguistic information. This approach to masking could implicitly allow a standard ASR system to guide the mask estimation process. In this chapter we define the alternative masking criterion and compare it to the IBM. While estimation results are not shown in this chapter, we do demonstrate that estimating the mask appears feasible and potentially simpler than estimating the IBM.[4]

5.1 Introduction

In Section 3.2 we introduced the concept of the IBM for speech enhancement. The previous chapter focused on how to use the IBM in ASR, but now we turn our focus to the estimation of the binary mask. Generally, binary mask estimation is used to improve one of two possible tasks, human speech recognition (HSR) or ASR. Regardless of the eventual goal, estimation techniques are similar.

Given that the IBM improves both HSR [99, 100] and ASR [31] performance, it would be reasonable to be agnostic of the eventual task. However, this ignores the potential benefits of the robust ASR task. If the goal is HSR, then you essentially have one opportunity

[4]Preliminary results were published in the *Proceedings of the 2012 IEEE International Conference on Acoustics, Speech, and Signal Processing* [32].

to produce a quality estimation. When working in robust ASR, you have access to both the recognizer and its output. While the recognizer cannot necessarily divulge the accuracy of the mask or recognition output, it can provide valuable information for refining the mask.

In this chapter, we present an alternative goal for binary mask estimation that relies on the linguistic content of the signal as opposed to the local SNR. Our proposed masking criterion is defined by the underlying linguistic content and is largely agnostic to the interfering noise. We demonstrate how this new mask is able to implicitly use information provided by an ASR system. Since this new masking criterion is inherently tied to ASR, we refer to it as an ASR-driven binary mask.

In Section 5.2, we outline previous approaches to mask estimation, including other top-down approaches. Our proposed ASR-driven binary mask is defined in Section 5.3. Specific details of our experimental setup are described in Section 5.4 and in Section 5.5 results comparing the ASR-driven binary mask to the IBM and results demonstrating the feasibility of estimating the ASR-driven binary mask are presented.. Section 5.6 presents conclusions and directions for future work.

5.2 Previous Work in Mask Estimation

Before discussing specific estimation techniques, we briefly review the IBM introduced in Section 3.2. Recall the equation for computing the IBM,

$$M(f,t) = \begin{cases} 1 & \frac{|S(f,t)|^2}{|N(f,t)|^2} > \theta \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

where f is a frequency band and t represents a particular time frame. $S(f,t)$ and $N(f,t)$ represent the amount of energy at a T-F unit for the clean speech and interfering noise respectively. If we assume the noise is additive, then the energy at any T-F unit for the mixed signal C is $C(f,t) = S(f,t) + N(f,t)$.

Given these two equations, we have three unknowns at each T-F unit, clean speech energy, interfering noise energy, and the instantaneous SNR. Of course, knowledge of any of the unknowns implicitly provides knowledge of the other two. Standard mask estimation techniques typically attempt to estimate one of these three unknowns or some combination. We review several techniques that utilize this low-level information for mask estimation. We also discuss previously proposed methods for utilizing top-down information.

5.2.1 Bottom-Up Approaches

One path to mask estimation is the estimation of the noise source. Spectral subtraction is a speech enhancement method that operates by first estimating the energy of the noise source [6]. The standard spectral subtraction algorithm then subtracts the estimated noise energy from the signal, but it can be used to compute a binary mask instead. The noise estimate itself is created by averaging the energy in the first few frames of the utterance. The performance of this technique works best when the implicit assumptions are met, stationary noise and initial silence.

Noise estimation techniques have evolved since this simple approach. More recent methods can track changes in noise energy to account for non-stationary noise sources [35]. However, in terms of mask estimation, the general technique is the same. Given an estimate of the noise energy at each T-F unit, the instantaneous SNR can be calculated and compared against a threshold. Here the accuracy of the binary mask is completely dependent on the quality of the noise estimation.

The spectral subtraction-based methods estimate the speech energy by subtracting the noise energy from the mixed signal. SNR is then calculated from these two estimates. Other techniques also incorporate general speech information [54]. Models of the clean

speech and noise-corrupted speech are compared. The masking decision is then based on the difference between how well the clean speech model and noise-corrupted model match the mixed signal. Here, the focus is on the speech and how the noise affects the clean speech characteristics. While this system produces an improvement over standard spectral subtraction based methods by using speech models, it does not incorporate higher-level linguistic information.

5.2.2 Top-Down Approaches

One of the first systems to couple the recognition and estimation process was that of Barker et al. [1], where the mask and speech were jointly decoded. They modify the standard ASR equation shown in Equation 2.2. Since they need to jointly decode the mask and the words, the process becomes more complex. For instance, the prior probability of the observed data is typically ignored, but it becomes dependent on the segmentation in their model and must be computed. Features are also an issue as spectral features are required in a missing data recognizer [13]. As we have already discussed, that experimental setup does not scale well to larger vocabulary tasks [83, 89]. The do-nothing approach of Chapter 4 would be difficult to incorporate as it relies on using normalized features and the normalization statistics would again be dependent on the segmentation.

However, the main issue with this system is that the search over possible binary mask labels is exponential. Barker et al. [1] attempt to alleviate this problem by first grouping T-F units. Labels are then assigned to the groups as opposed to the individual T-F units. This approach of grouping and then labeling is common among CASA-based approaches [40, 97]. Traditionally, these groupings are determined by low-level features [92] and is why we consider this a joint bottom-up top-down approach. While this system has obvious

problems, as we have outlined, it was a novel approach and one of the first to address the problem of including linguistic information in the mask estimation task.

A more recent study by Srinivasan and Wang [92] presents an alternative approach to integrating bottom-up and top-down information. They first use a spectral subtraction algorithm to generate a progressive mask. A progressive mask seeks to remove as much of the noise dominant T-F units as possible. While much of the speech-dominant units may be removed too, the goal is simply to remove the influence of noise information as much as possible. Once the progressive mask is computed, recognition is performed using an HMM and lattices are obtained.

At this point, a new type of mask is computed from the original signal. It is essentially the opposite of the progressive mask; only T-F units that are highly likely to be noise-dominant are masked. The remaining T-F units are considered to be unknown. Now the lattices are jointly decoded while the mask value is determined for the unknown units. Their values are determined by comparing the values of the features to the Gaussian in the HMM. In the end, both a mask and a recognition output is produced.

This system partially alleviates some of the issues in Barker et al. [1]. The prior probability of the data no longer needs to be modeled and the evaluation of all possible segmentations is avoided. However, since the limitation of spectral features still remains, the extension of these results to larger vocabulary tasks is also unlikely. In fact, even on the TIDigits [59] task presented in [92], performance is not as strong as simply using standard unenhanced cepstral features. It appears that in order to utilize top-down information for larger vocabulary tasks, the requirement of spectral features for recognition must be relaxed. In the remainder of this chapter, we present an alternative approach to utilizing linguistic information that will allow for the use of cepstral features.

5.3 The ASR-Driven Binary Mask

The IBM chooses which T-F units to mask based on the instantaneous SNR. We propose an alternative criterion for masking T-F units based solely on the underlying linguistic content. Our hypothesis is that any T-F unit which should contain large amounts of speech energy should remain unmasked regardless of the amount of noise energy. Also, it should be safe to mask any T-F unit which should contain very little speech energy even if the amount of noise energy is much smaller. Motivation for our hypothesis stems from the work of [99] where they showed human subjects could understand binary masked noise signals. An IBM was first calculated from a mixture of noise and speech. A noise signal containing no speech information was then masked with the computed IBM. Even without any underlying speech energy in the signal, listeners perceived recognizable speech simply from the patterns of energy imposed by the binary mask. In order to use our proposed masking criterion, we require three pieces of information.

1. The underlying linguistic information at every time frame of a given speech signal.

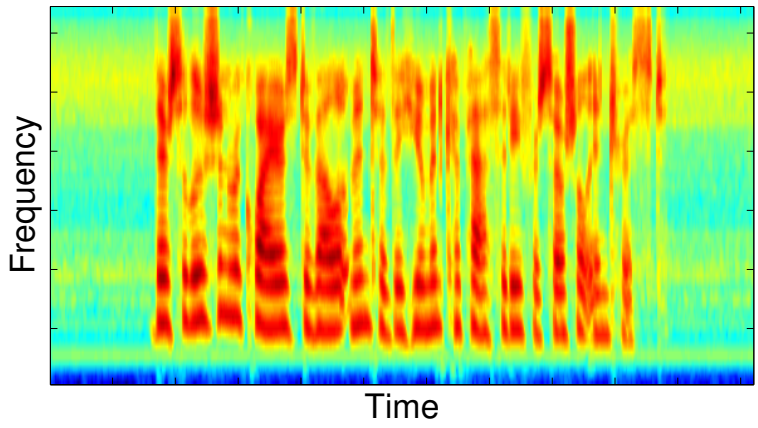
2. The expected amount of energy at each T-F unit for a given linguistic element.

3. A threshold the expected amount of energy must surpass in order to remain unmasked.

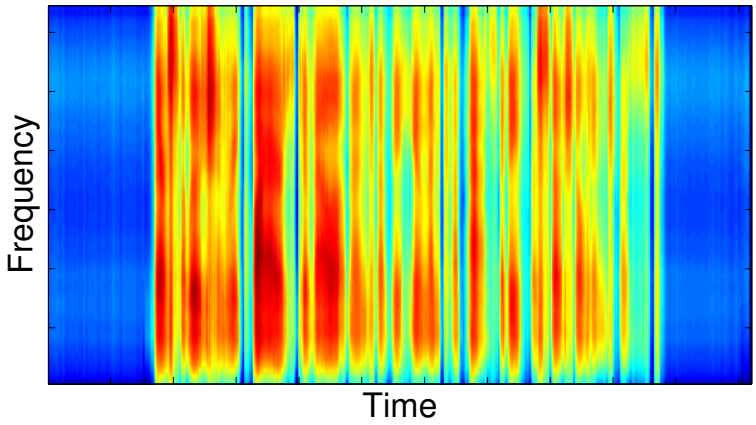
Since our focus is on ASR, we can utilize the linguistic units modeled by the recognizer. The standard HMM recognizer, as described in Chapter 2, utilizes multi-state triphone units. By force-aligning the correct word sequence to a clean speech signal, we can obtain a labeling of one triphone state at each time frame. We consider this labeling to be a description of the underlying linguistic information in the signal.

Now that we know the underlying linguistic information, we need to know some information about the expected energy in each T-F unit for that linguistic element. For this thesis, we represent the signal in the time-frequency domain as a cochleagram with 64 frequency channels. We described the computation of the cochleagram in Section 3.3.2. We use a simple model of the expected energy. For every utterance in a training set, we force-align the speech to the correct word sequence to provide a triphone state label for each frame. The amount of energy at each T-F unit is represented as the cube root of the value in the cochleagram. The cube root is used to simply decrease the dynamic range. Next, each T-F unit is normalized by the total energy in the frame since we are more interested in the distribution of energy than the actual energy itself. The model for each triphone state is simply the mean of all the energy ratios with the same label.

To better visualize these prior models on the ratio of the cube root energy, we can recreate a speech signal from an alignment and the total energy at each time frame. Since the models only contain energy distribution information, we need to know the total energy in each frame to obtain something similar to the original speech signal. Again, this is simply for the purpose of visualization. Figure 5.1 shows a cochleagram of an utterance followed by a recreation of that utterance using our prior models. The gross structure of the cochleagrams are similar, but the artificially constructed cochleagram is smoother and lacks the finer formant structure and speaker specific details of the original. This is to be expected as the models are averaged over all instances. More complicated mixture models with higher order statistics might be able to better represent the energy as each subphonetic state would be represented by multiple models, but they would be more difficult to use. We will first determine the efficacy of our simple prior models for masking.



(a) Cochleagram



(b) Prior Model

Figure 5.1: Comparison of (a) Cochleagram and (b)Prior model representations of the same utterance.

The final piece of information required is a threshold to compare against the prior models. The threshold is split into two pieces where the first piece is termed the background prior. The background prior can be viewed as the prior on the expected distribution of energy given no other information. While this prior could be determined based on corpus specific characteristics, we found a prior with equal energy at each frequency worked well. Using the background prior alone as a threshold does not work well; it masks high energy speech components too much and low energy speech components too little. Instead we combine it with a second piece of information, the ratio of average frame energy to energy in a given frame. The assumption is that frames with little energy likely contain little speech energy and should be masked more strongly than frames with high energy.

Now that we have conceptually defined our masking criterion, we can more formally define it as

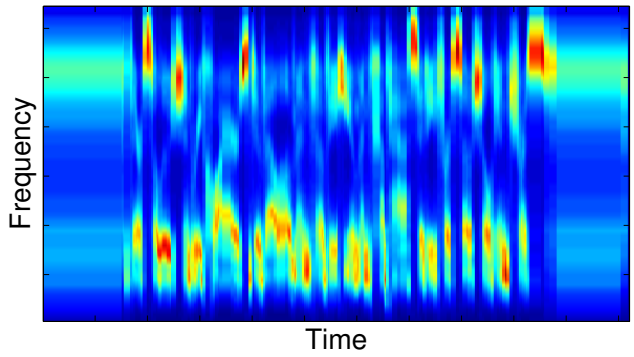
$$M(f,t) = \begin{cases} 1 & \alpha_{f,s_t} > \beta_f r_t^{\gamma} \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

where r_t is defined as

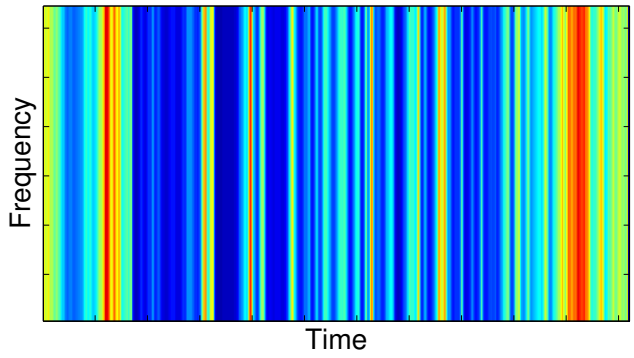
$$r_t = \frac{\text{average frame energy}}{\text{frame energy in frame } t} \tag{5.3}$$

and t is the time frame, f is the frequency band, α_{f,s_t} is the relative spectral energy in frequency f for the prior vector associated with HMM triphone state s_t, β_f is the background prior, and γ is a factor used for nonlinearity.

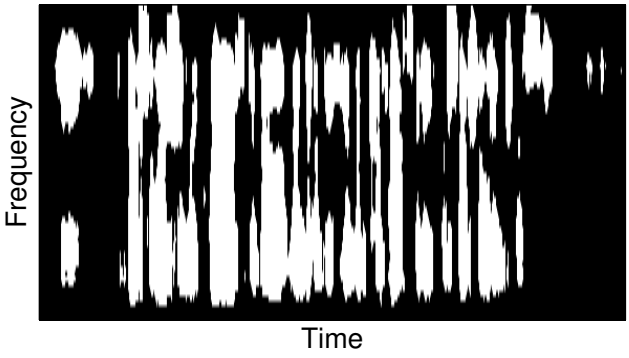
The masking criterion seeks to keep any T-F units that should have strong energy based on the speech segment that we are observing. This is accomplished by seeing if the energy percentage is greater for the speech prior than the weighted background prior. Weighting the background prior modifies the prior based on the amount of energy in a frame. We assume frames containing more energy are more likely to have strong speech energy also,



(a) Acoustic Prior Model



(b) Background Prior Model



(c) ASR-Driven Binary Mask

Figure 5.2: Example ASR-Driven Binary Mask process. (a) Acoustic prior model. (b) Background prior model. (c) ASR-Driven Binary Mask generated by comparing the acoustic prior model and background prior model.

though we recognize this assumption could be incorrect when the noise is concentrated in a short time window or impulsive. The γ value makes the weighting nonlinear so frames which deviate greatly from the mean are affected more. In this study a value of 2.5 is used, but different values have little effect on performance. An example can be seen in Figure 5.2 where the left and right half of the inequality is represented visually and the final mask is displayed.

We want to emphasize that once the state model is chosen, the ASR-driven mask is based solely on the expectation of which T-F units should have strong speech energy and the relative energy of the frame. No assumptions about noise are made and the individual T-F units of the data have no bearing on the mask estimation process. Work in human speech intelligibility has also shown that the IBM pattern is more important than the local SNR at each T-F units [61], supporting this approach. Once the mask has been computed, we multiply it by the original signal and resynthesize the waveform. Standard ASR features can then be calculated from the enhanced waveform. As with the IBM, we have defined how to compute an oracle mask. The question of how to estimate the mask remains. We will examine that question in detail in the remainder of this thesis, however, the remainder of this chapter will focus on the performance of the oracle ASR-driven mask compared to the IBM and demonstrating the feasibility of estimating the ASR-driven mask.

5.4 Experimental Setup

We use the HMM toolkit (HTK) [105] for our recognition system. The acoustic model consists of intra-word triphones; each triphone has three states, modeled by a mixture of 16 Guassians per state. A bigram language model is used during decoding. The CMU dictionary was used for our pronunciation dictionary. Our 39 dimensional feature vector

is comprised of mean and variance normalized PLP features, including the delta and acceleration coefficients. While most systems using binary masking for speech enhancement use some type of missing feature recognition system [13, 83], we have found this can be unnecessary as long as the features are variance normalized [31].

All evaluations are performed on the Aurora4 corpus [75], a 5000 word closed vocabulary task. This task is a modification of the Wall Street Journal (WSJ0) database where noise has been added to the clean speech recordings at various SNR. Approximately 21K prior energy distribution models were computed from the training set.

Our experiments in this chapter use oracle masks. For an IBM, the oracle information required is the instantaneous SNR, which can be computed by comparing the clean speech utterance to the mixture in each of the test cases. For the ASR-driven mask, we need to know the true triphone state at each frame. We obtain this information by force-aligning the clean speech utterance with the correct word sequence. In the next section, we evaluate the oracle masks using the setup just described.

5.5 Oracle Mask Results

We will evaluate our new mask in two separate manners. First, we will directly compare it to results using the IBM. Then we will examine the feasibility of estimating the mask. More specifically, we will explore methods of reducing the total number of prior models to consider and whether all 21K prior models are necessary.

5.5.1 Oracle Mask Comparisons

Table 5.1 present word error rates for the two oracle masks. The *Baseline* result uses no enhancement and provides a floor for performance of any enhancement technique. The IBM result provides a ceiling for performance since any other binary mask is unlikely to

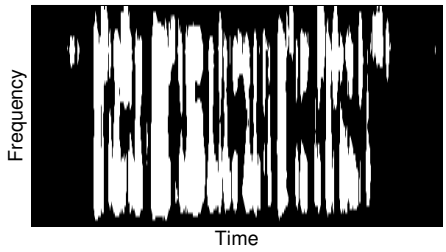
Mask Type	car	babble	rest.	street	airport	train	avg
Baseline	27.7%	34.7%	37.3%	39.9%	35.5%	42.2%	36.0%
Clean Speech Oracle	18.4%	20.0%	23.7%	20.6%	22.7%	21.7%	21.2%
Ideal Binary Mask	17.6%	16.8%	15.2%	18.1%	15.6%	19.1%	17.1%

Table 5.1: Word error rates for oracle mask types on the Aurora4 dataset. Comparison of the ideal binary mask and ASR-driven binary mask. Word Error Rate for clean speech is 9.8%

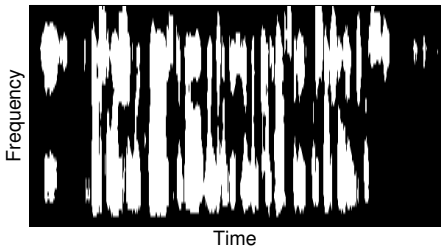
perform better. Our ASR-driven oracle mask is labeled as *Clean Speech Oracle*. We use this term as the oracle information comes from aligning the clean speech. While it does not perform as well as the IBM, it still provides a 41% error reduction over the baseline.

In Figure 5.3, we show a comparison of the two oracle masks on clean speech mixed with factory noise at an SNR of 10 dB and 5 dB. While the masks are similar, there are some obvious differences. Since the ASR-driven mask ignores the noise and the energy in individual T-F units, it is more stable to variations in SNR. The IBM becomes progressively sparser as the SNR decreases. This phenomenon of the the IBM certainly is not a flaw, but the ASR-driven mask may be easier to estimate since the mask is more stable. Another major difference is in the formant structure; as the SNR decreases, the formant structure is much more apparent in the traditional IBM.

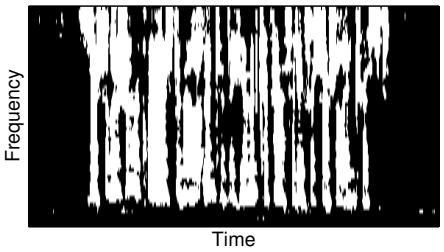
Clearly, if a perfect IBM estimator existed, its performance would be superior to the ASR-driven mask. However, both masks produce significant improvements over an unenhanced baseline. The true test of these two masks is not in their oracle performance, but in the performance in methods used to estimate them. In the next section, we examine the feasibility of estimating the ASR-driven mask.



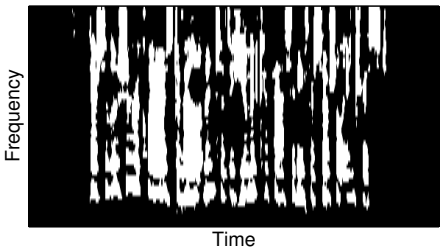
(a) ASR-Driven Mask for 10 dB Factory Noise



(b) ASR-Driven Mask for 5 dB Factory Noise



(c) Ideal Binary Mask for 10 dB Factory Noise



(d) Ideal Binary Mask for 5 dB Factory Noise

Figure 5.3: Comparisons of the oracle ASR-Driven Binary Mask and the Ideal Binary Mask on a clean speech utterance mixed with factory noise at 10 and 5 dB. The ASR-Driven mask is smoother and more stable. The IBM varies with changes to the noise source and masks spectral valleys between formants and harmonics. (a) Ideal Binary Mask for 10 dB Factory Noise. (a) Ideal Binary Mask for 5 dB Factory Noise. (c) ASR-Driven Binary Mask for 10 dB Factory Noise. (d) ASR-Driven Binary Mask for 5 dB Factory Noise.

5.5.2 Feasibility of ASR-Driven Mask Estimation

We demonstrate the feasibility of estimating the ASR-driven binary mask in two ways. First, we will show that ASR results can be used to provide suitable candidate models to choose from. The usability of our mask rests on the assumption that it is easier to estimate than the IBM. Given that estimating one frame of the mask rests on choosing from one of roughly 21,000 models, this assumption obviously requires testing. We partially test this hypothesis by examining the result of reducing the total number of models used.

In order for our mask to be considered an ASR-driven mask, the ASR process must assist in the binary mask estimation. We believe the results from a baseline recognition system can reduce the number of candidate models to consider. The two standard ways of receiving results from a recognizer are as a lattice and as an n-best list, both of which were described in Section 2.4. The typical output of a recognizer could simply be considered a 1-best list. We consider these two types of output to be at opposite extremes where the lattice will produce many candidates per frame and the n-best list, depending on the size of n, will produce far fewer candidates.

Given a lattice, we can collect every possible state hypothesized at each frame. For our Aurora4 dataset, this is an average of 8.7 states per frame, a significant reduction from the 21,000 possible states when given no other information. The question is how much has this reduction in candidate models cost in terms of potential accuracy in the final recognition. We can answer that question by generating a binary mask using only this reduced set of candidate models.

In the previous chapter we generated an oracle mask by using the model that corresponded to a clean speech alignment. To generate a mask from the reduced set of models, we select the model that most closely resembles the true model. Once again, this produces

Mask Type	car	babble	rest.	street	airport	train	avg
Baseline	27.7%	34.7%	37.3%	39.9%	35.5%	42.2%	36.0%
100-Best Oracle	22.8%	28.8%	31.8%	31.7%	29.6%	31.2%	29.3%
Lattice Oracle	20.7%	24.2%	27.3%	24.9%	26.1%	26.5%	24.9%
Clean Speech Oracle	18.4%	20.0%	23.7%	20.6%	22.7%	21.7%	21.2%

Table 5.2: Word error rates for oracle mask types on the Aurora4 dataset. Comparison of oracle ASR-driven binary masks with reduced candidates generated from a baseline ASR system.

an oracle mask. Results can be seen in Table 5.2. *Clean Speech Oracle* refers to the oracle ASR-Driven mask produced from using the models in the clean speech alignment. *Lattice Oracle* refers to the mask we just described. Clearly, the reduction in candidate models has also cost some amount of potential accuracy. However, the mask still provides an average decrease in WER of 31% over the unenhanced baseline.

Now we consider the n-best list. For our experiments here, we will use an n of size 100. For each sentence in the 100-best list, we force align the sentence to the unenhanced speech signal. The alignment produces a hypothesized state at each time frame. Since the sentences in a 100-best list are typically very similar, we do not end up with 100 potential states at each frame, but a much reduced 1.7 average states per frame. Once again we see a large reduction in candidate states for each frame. Again, we must answer the question of how much this reduction in candidate models costs in terms of potential recognition accuracy.

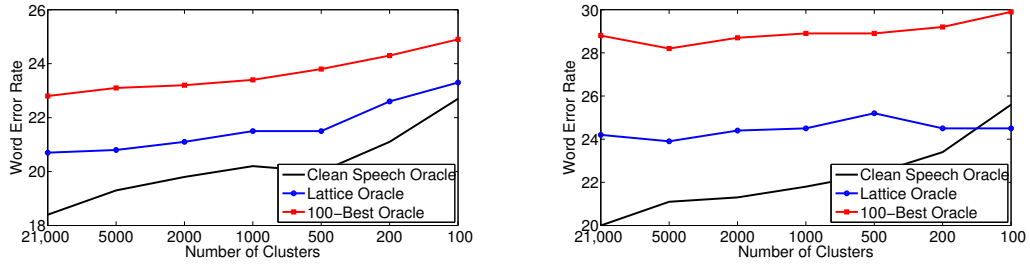
Results are shown in Table 5.2. Our new mask is created in the same manner as *Lattice Oracle*. The results for using the 100-best list to reduce the candidate models is labeled *100-Best Oracle*. We see another increase in WER compared to both *Clean Speech Oracle* and *Lattice Oracle*. However, it still produces an average decrease in WER of 19% over

the unenhanced baseline. We have shown that an ASR system can help constrain the mask estimation process, but at the cost of potential maximum recognition accuracy. In the next chapter we will explore whether the drop in potential accuracy is worth the reduction in total models considered at each frame.

The second question we would like to analyze is the specificity required in these models. Do we truly require the full set of 21,000 models for our system? We explore this question by clustering the prior models to create a smaller set of models. By treating each model as a 64 dimensional vector, we can use K-means [74] to cluster. For our experiments we tried a range of clusters between 100 and 5000. When a set of models are clustered together, the clustered model simply becomes the mean of the representative set. This was a very simple approach to the clustering problem. Given the additional information implicit in our models, a more complicated process could be used that leverages the phonetic similarity between models and also the relative frequency of the models. However, our simple approach worked well for our study.

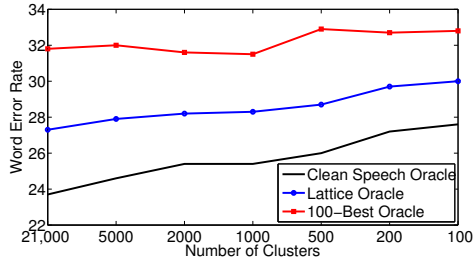
Results can be seen in Figure 5.4; an average over all test conditions can be seen in Figure 5.5. Each plot corresponds to one of the test conditions in the Aurora4 test set. The y-axis is WER and the x-axis shows the number of models used. As with the previous results *Clean Speech Oracle* refers to selecting the best model from all possible models, *Lattice Oracle* refers to selecting the best model from the set of hypotheses generated by the lattice, and *100-Best Oracle* refers to selecting the best model from the set of hypotheses generated by the 100-best list. We should also mention every result is significantly better than results using the unenhanced baseline.

Several patterns are immediately visible. Every system sees a steady increase in WER as the number of models decreases. Also, we clearly see that more candidate models always

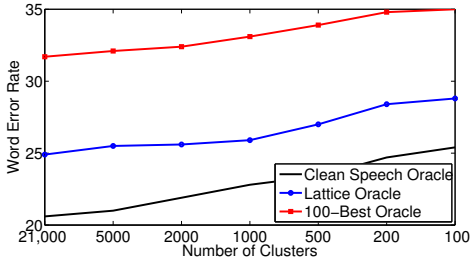


(a) Car noise

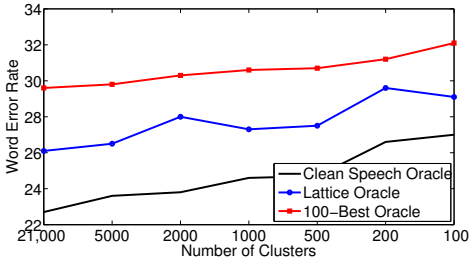
(b) Babble noise



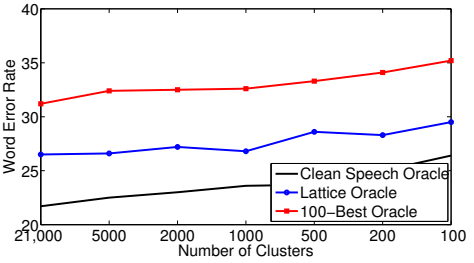
(c) Restaurant noise



(d) Street noise



(e) Airport noise



(f) Train noise

Figure 5.4: Word error rates for Aurora4 using clustered models. The masks are oracle ASR-driven binary masks where the candidate models are chosen from all models, lattice, or 100-best list. (a) Car noise. (b) Babble noise. (c) Restaurant noise. (d) Street noise. (e) Airport noise. (f) Train noise.

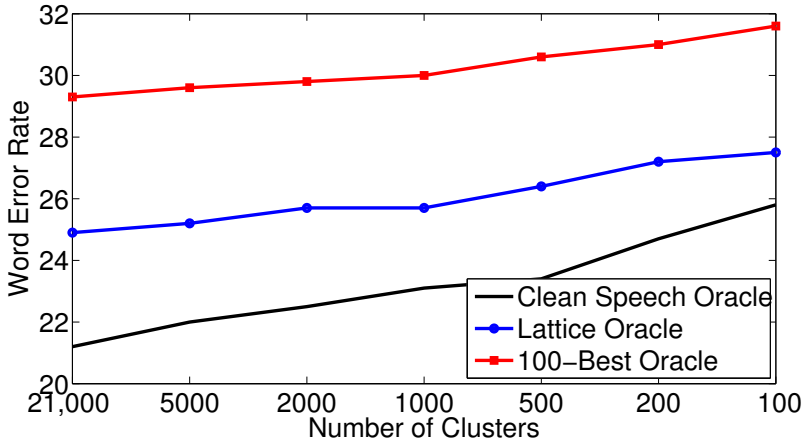


Figure 5.5: Word error rates averaged over all conditions for Aurora4 using clustered models. The masks are oracle ASR-driven binary masks where the candidate models are chosen from all models, lattice, or 100-best list.

produce a better result when the number of clusters is the same. In fact, it isn't until the number of clusters is reduced to 200 that we begin to see similar results between *Clean Speech Oracle* and *Lattice Oracle*. Even with only 100 clusters, *Lattice Oracle* performs better than *100-Best Oracle* when using all models. Of course, none of these patterns are surprising. Results are similar across all noise types; car noise differs somewhat as performance is better in that task than the other noise types.

Based on these results, we have established the feasibility of estimating the ASR-driven binary mask in two ways. We have shown that we can obtain valid candidate models based on the hypotheses provided by an ASR system recognizing unenhanced speech. However, reducing the list of candidate models comes at a cost to potential accuracy. Obviously estimating the mask will be a balancing act between the increased potential from more candidates and degraded performance caused by the inability of the estimation algorithm

to identify the best model from and increased set of models. We will examine that trade-off in future chapters.

We also showed we could simply reduce the total number of prior models through clustering. Again, this comes at a cost in potential performance, but demonstrates how sensitive the results are to the specificity of the models. A reduction in the total number of prior models by 90% only produces a small increase in WER. Since it is possible to use these clustered models, during estimation it may be possible to consider using a clustered model rather than selecting from a small number of candidate models.

5.6 Conclusions

Most mask estimation algorithms make little use of linguistic information in the mask estimation process. Some algorithms are so general they do not leverage any information about the source. Of the systems which do attempt to incorporate knowledge about speech into the process, the focus tends to be only the most general characteristics of speech. While a few systems have tried to incorporate top-down linguistic information, they generally suffer from the use of spectral features and the complexity of decoding over every possible mask pattern.

We hypothesize that one reason systems ignore any available linguistic information is due to the definition of the IBM; it is only concerned with the energy of the source and interference at the T-F unit level. We propose an alternative masking criterion that is defined based on the underlying linguistic information. When given oracle information, our criterion does not perform as well as the IBM, but it may be easier to estimate. We note that our mask definition is just one way of defining an ASR-driven mask; alternative masks using other types of information or even segmental structure could be used. We have

established that hypotheses from a baseline ASR system can be used to guide the estimation of the ASR-driven binary mask. We have also provided some evidence for the feasibility of estimating the mask.

Now that we have established a straightforward and simple method for incorporating linguistic information into the mask definition, we need to examine how exactly to estimate the mask. Due to the manner in which we have defined the mask, we have altered the mask estimation process such that it now relies on being able to select from a few candidate models at each time frame. In the next chapter we will propose a method for estimating the ASR-driven binary mask.

CHAPTER 6: ESTIMATING THE ASR-DRIVEN BINARY MASK

The previously defined ASR-driven binary mask is an alternative masking criterion that explicitly incorporates knowledge about the underlying linguistic information into the mask definition. Estimating the IBM can be thought of as a binary classification task where a decision is made at each T-F unit. Due to the way the ASR-driven binary mask is defined, it alters the estimation process to be a multi-class classification task, or model selection task, at the frame level. In this chapter we will explore methods for estimating the binary mask through a model selection process and also investigate the best way to incorporate the estimated mask into the recognition task.[5]

6.1 Introduction

In the previous chapter, we introduced the ASR-driven binary mask. The mask associates a simple masking model to a set of subphonetic models corresponding to the underlying acoustic model representation in an ASR system. This alters the mask estimation process to be a model selection task where a specific model must be chosen at the frame level. In general, this is likely to be a very difficult task as the number of potential models is very large. In our experiments, the number of models exceeds 20,000. However, we demonstrated in the previous chapter that the number of potential models at each frame could be reduced to very few by incorporating the hypotheses from an ASR system with

[5]Preliminary results were published in the *Proceedings of the 2012 IEEE International Conference on Acoustics, Speech, and Signal Processing* [32] and the improved model selection framework was submitted to the *2012 Interspeech Conference*.

only a small cost in potential accuracy. Our ultimate goal in this chapter is to propose a method for selecting the best model from a set of hypothesized candidate models.

Prior to discussing the estimation process, we will first discuss the application of the estimated mask in ASR. In Chapter 4 we discussed the application of the IBM in ASR and determined that missing data techniques were either unnecessary or could potentially only offer small improvements in accuracy. However, we have discovered that some amount of compensation may be necessary when using partial or estimated masks. In Section 6.2 we discuss this issue in detail.

In Section 6.3 we discuss the simple estimation method used in our pilot study [32]. We examine baseline techniques in Section 6.4. We also demonstrate that as the accuracy of the baseline estimation improves, we can also use the baseline technique to improve our estimation of the ASR-driven binary mask. Our improved model selection algorithm based on a discriminatively trained linear chain sequence model is presented in Section 6.5. Conclusions and possible avenues for future work are presented in Section 6.6.

6.2 Applying Partial and Estimated Masks

In Chapter 4 we demonstrated that strong recognition performance could be obtained by directly using the IBM without the use of any compensation from missing data ASR. However, we have discovered that there are situations where some type of compensation must be made during feature calculation. When using partial masks and estimated ASR-driven binary masks, we were unable to achieve an improvement over an unenhanced baseline system using standard feature calculation. As we will show, the reason is tied directly to the variance normalization that allowed the IBM to be used without compensation.

We explain these surprising results and investigate methods to compensate for the issues discovered. First, we will examine the case where we have a partial mask. A partial mask is any mask that only attempts to mask a subset of T-F units and leaves other regions unmasked regardless of the underlying interference. While we make little use of partial masks in this thesis, they help to easily illustrate the issues we face and parallel the manner in which we discovered them.

We will also explore the issues surrounding the use of estimated masks. Since we are estimating the ASR-driven binary mask, we are already dealing with masks which are much different than the IBM investigated in Chapter 4. Errors in the estimation will further deviate from the IBM. It is possible the ASR-driven binary mask is more sensitive to errors. When dealing with estimated masks we utilize a different method for compensation than when dealing with partial masks.

6.2.1 Partial Masks

A partial mask is any mask that only operates on a subset of T-F units. Our focus will be on partial masks that make their decision at the frame level. Partial frequency masks could also be used, but we will show their use is more complicated in this framework. Given that our approach to mask estimation requires the selection of one candidate from a set for each frame, it follows that some choices will be easier than others. There is a large variation in the number of candidates per frame; some frames contain only one or two possible candidates, while others will have dozens. A partial mask would be an obvious tool as we could simply operate on the frames we are confident in. However, we discovered through informal tests that partial masks did not improve recognition results. Surprisingly, even a partial IBM only resulted in decreasing performance. We had expected a partial

Mask Type	factory 10 dB	factory 5 dB
Baseline	26.3%	51.3%
Partial Mask	37.7%	64.1%
Ideal Binary Mask	13.5%	19.5%

Table 6.1: Word error rates for WSJ0 speech mixed with factory noise. Baseline refers to unenhanced speech. Partial Mask masks the first half of the utterance with the IBM and Ideal Binary Mask uses the full IBM. Results demonstrate that a partial binary mask does not improve ASR performance.

IBM to produce a result between the complete IBM and the unenhanced speech. Since the result did not fit with our expectation, it warranted further exploration.

For illustrative purposes, we will consider a partial binary mask where the first half of each utterance is properly masked and the second half is unenhanced. While a mask of this type would be uncommon in practice, it does provide a simple example to study. The partial mask is used in the same way as in the previous chapter; the signal is masked in the cochleagram domain and resynthesized to the time domain where our standard features are calculated. We will show the results of a few simple experiments. Our experimental setup is nearly identical to that of the last chapter, except that we are using a development dataset; since we are exploring a variety of tests and methods, we do not want to lose the potential for refining any of these methods prior to evaluating performance on the test set.

In Table 6.1 we see results on utterances from WSJ0 mixed with factory noise from the Noisex92 database [95] at 10 dB and 5dB SNR. *Baseline* refers to recognition of the unenhanced speech, *Partial Mask* is the mask we just described, and *Ideal Binary Mask* is the oracle mask. The expectation is that the *Baseline* result would set a floor for performance of the partial mask and *Ideal Binary Mask* would set the ceiling. Since the *Partial Mask* is just a combination of the other two results, it is reasonable to expect performance to fall

Mask Type	factory 10 dB	factory 5 dB
Baseline	26.3%	51.3%
Dual Normalized Partial Mask	21.8%	37.3%
Ideal Binary Mask	13.5%	19.5%

Table 6.2: Word error rates for WSJ0 speech mixed with factory noise. Baseline refers to unenhanced speech. Dual Normalized Partial Mask masks the first half of the utterance with the IBM and normalization statistics have been calculated separately for both the masked and unmasked halves of the utterance. Ideal Binary Mask uses the full IBM. Results illustrate the need to separately calculate normalization statistics when using partial binary masks.

about halfway between the other two results. However, the *Partial Mask* performs worse than the *Baseline*.

We have once again encountered a situation where applying a binary mask actually harms performance. This is similar to the phenomenon seen in Chapter 4 where we identified why research in missing-data ASR ignored the direct use of the IBM without any form of compensation. In that case, we were able to demonstrate as long as the final features were mean and variance normalized, the IBM could be used without any type of missing-data compensation. Since we are still performing variance normalization, why does the partial mask fail to provide any benefit?

When performing data normalization, there is an implicit assumption that all of the data comes from the same distribution. By using a partial mask, we have violated this assumption. The partial mask has created a situation where the first half of the utterance represents masked noisy speech and the second half represents unenhanced noisy speech. If we calculate normalization statistics over the entire utterance, obviously we have a mismatch.

We can test this hypothesis by modifying the way we calculate the normalization statistics. Instead of calculating statistics over the entire utterance, we separate the utterance

into two parts based on whether or not each frame falls under the purview of the partial mask. Given the two sets of statistics, we can normalize the features from the two halves of the utterance separately. Table 6.2 presents results using this dual normalization. We can clearly see that when using the dual normalization, the results fall between the *Baseline* and the *Ideal Binary Mask* as we had originally expected.

We have identified a problem and provided a solution, but this likely applies only to this very specific type of partial mask. The dual normalization is a valid approach because our example partial mask divides the utterance in half; the underlying distribution of speech is likely to be very similar for both halves. A more realistic partial mask would not maintain this even distribution because certain types of speech would be masked with higher confidence.

Recall the previous chapter where we described a method for selecting a small number of candidates for the ASR-driven binary mask by using the hypotheses from a first pass ASR lattice. We will use this approach to generate a more realistic partial mask for analysis. Our partial mask will choose to first mask those frames which incur the least risk. For our purposes, risk refers to the maximum number of T-F units that can be classified incorrectly in a given frame, assuming the correct mask is in the set of hypotheses. Our assumption is that only masking frames with little risk will decrease the penalty of an incorrect decision. Since we have not discussed any estimation methods yet, we will assume the partial mask uses the mask that most closely matches the oracle ASR-driven binary mask in every frame.

Table 6.3 shows results using this partial mask where the risk at any masked frame is less than 5. Again, *Baseline* refers to the unenhanced speech. *Single Normalization* refers to using a single set of normalization statistics calculated over the entire partially masked utterance. Our proposed method of using two sets of normalization statistics is labelled

Mask Type	factory 10 dB	factory 5 dB
Baseline	26.3%	51.3%
Single Normalization	26.7%	51.9%
Dual Normalization	26.1%	49.8%
Global Normalization	23.6%	45.2%
Ideal Binary Mask	17.3%	23.1%

Table 6.3: Word error rates for WSJ0 speech mixed with factory noise. Baseline refers to unenhanced speech and ASR-Driven Binary Mask uses the oracle ASR-driven binary mask . The three other results utilize the oracle ASR-driven binary mask in frames where the risk < 5, corresponding to approximately 50% of frames in the 10 dB case and 33% in the 5 dB case. Single normalization uses a single set of statistics calculated over the entire utterance. Dual normalization uses statistics calculated separately for masked and unmasked regions in the utterance. Global normalization uses unenhanced features for unmasked regions and statistics calculated over a global training set for masked regions.

Dual Normalization, but it does not perform well. The most likely cause is that the data is now unbalanced; a large portion of the masked data will be silence regions.

We now need a method for calculating normalization statistics when the data in the utterance is unbalanced. Instead of calculating statistics for each utterance, we can calculate the statistics over a set of utterances where we know they have either been fully masked or the partial masks used represents an even distribution of the data. We take our standard training set and mix it with white noise at a random SNR between 0 and 20 dB. Each utterance is masked using the oracle ASR-driven binary mask. Normalization statistics are calculated over the entire training set.

At test time, any frames associated with the partial mask are normalized using the statistics from the training set. We have two choices for normalizing the unenhanced frames. Since we still have the original unenhanced data, we could use the complete unenhanced

utterance to calculate statistics or we could simply use the features calculated from the unenhanced data. In our experiments, we simply replace the unmasked frames with feature calculated from the unenhanced utterance.

Results can be seen in Table 6.3. Using the *Global Normalization*, we see an improvement for the partial mask we have discussed. It does not work as well as dual normalization in the case where the utterance is divided in half. This means that if we could calculate the true underlying statistics, then we should expect better performance. However, global normalization is a valid alternative in lieu of the local statistics.

We have presented a solution to the problem of applying partial binary masks when using mean and variance feature normalization. If we do not know the true local statistics, we can use global statistics calculated over a training set. Our proposed approach deals only with partial masks at the frame level. If the partial mask is defined in terms of frequencies or is not consistent across all frequencies in a particular frame, then this approach will not work. As the cepstral transformation incorporates information from all frequencies in every feature, altering the normalization at the frequency level will not help. Since we do not use partial frequency masks in this thesis, we will not consider them any further. However, we do use estimated masks, which will be similar in some respects to a partial frequency mask. In the next section we discuss estimated or incorrect binary masks.

6.2.2 Estimated Masks

In the previous section, we discussed the issue of applying partial binary masks. Using estimated masks present another problem. The binary mask will contain errors and will make incorrect decisions at certain T-F units. If the mask makes more correct decisions

than incorrect decisions, then it is reasonable to assume it should still provide some improvement. We will show that estimated masks tend to decrease performance compared to an unenhanced baseline. The cause is likely the same as with partial masks; the normalization statistics do not match the data. We demonstrate that by relaxing the hard constraint of zero energy in masked regions, we can offset the errors in the mask estimation and improve overall performance.

Our experiments will be performed on the same set of data as in the previous section. For these experiments, we require an estimated mask. We will use a very simple method for estimating the ASR-driven binary mask. The details of the mask estimation process are not important for this discussion, but will be described in the next section where we introduce a baseline framework for estimating the ASR-driven binary mask.

We believe the poor performance of estimated masks is still related to the calculation of normalization statistics. With the partial mask, we were dealing with two types of data. In this case, we have a wide variety of data related to the accuracy of the mask at a given frame. We have silence regions that are unmasked, silence regions completely masked, correctly masked speech, and frames that mask the speech and leave noise-dominant regions unmasked. This mixture of data obviously does not come from a single distribution. We attempt to combat this issue by modifying the percentage of energy left in masked T-F units.

Traditionally, a binary mask assigns a value of unity to speech-dominant units and zero to noise-dominant units. When the mask is applied to the utterance, masked T-F units will have no energy. We mentioned in Chapter 5 that we obtained better performance through the use of a small noise floor. Instead of using values of 1 and 0 in the mask, we used 1

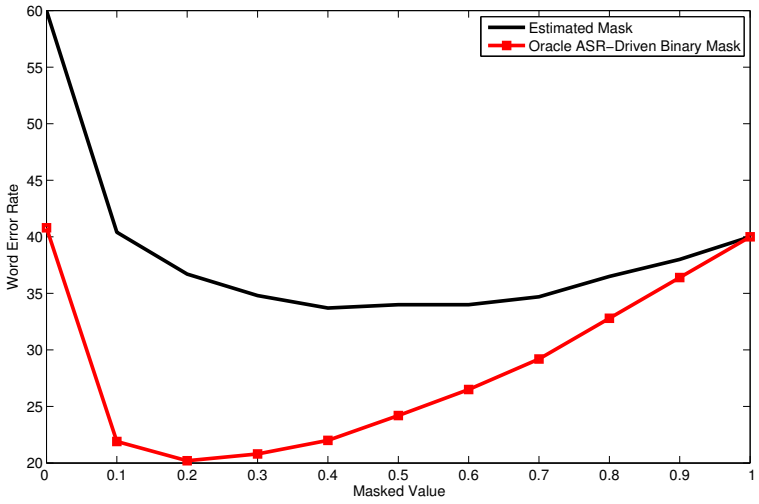


Figure 6.1: Word error rates averaged over the WSJ0 development set mixed with factory and babble noise at 10 dB and 5 dB SNR. Results for both an estimated and oracle mask are shown for various values for masked T-F units.

and 0.1. Other values for the masked regions provided similar results. We use the same approach for dealing with estimated masks.

Results averaged over the WSJ0 development set mixed with babble and factory noise at 5 and 10 dB SNR can be seen in Figure 6.1. Various values for the masked units are tested between 0 and 1. The optimal value for the masked regions vary depending on the test condition, but a value of 0.5 seems reasonable. Based on these experiments, we will use a value of 0.5 for all future experiments in this thesis.

One obvious question is what happens as the estimated mask becomes more accurate. We can examine this question by evaluating the results of varying the mask value when using an oracle mask. Results for the oracle ASR-driven binary mask can also be seen in Figure 6.1. A sharp decrease in WER is seen when the masked value increases above 0. A steady increase is then seen as the masked value increases. The overall change in

performance is small when comparing the best performing masked value and the value of 0.5.

Based on these results, we have shown that using a nonzero masked value can improve performance when using estimated binary masks. While performance is better for oracle masks when using smaller values, the difference is not great. We note that it may be possible to slightly improve performance by modifying the masked value based on the test condition, but we will not explore this task in our experiments. Altering the masked value is a very simple and effective method for improving the performance of estimated binary masks.

6.3 Model Selection for ASR-Driven Binary Mask

Chapter 5 presented an alternative criterion for binary masks termed the ASR-driven binary mask. The mask is defined based on the underlying linguistic information in the speech signal. T-F units which typically contain large amounts of speech information are unmasked while units that usually contain small amounts of speech information are masked. The masking decision is made without any consideration of the underlying characteristics of the interfering noise. In that chapter we also demonstrated how the noisy output of an ASR system could be used to guide the mask estimation process.

If the ASR system returns more than a 1-best hypothesis, a set of possible subphonetic candidates can be obtained for each frame. These subphonetic candidates correspond to the acoustic models in the HMM-based recognizer. Each subphonetic unit has a simple model associated with it which, in conjunction with information about the average energy distribution in the utterance, determines the masking pattern for a given frame. Given the

Mask Type	car	babble	rest.	street	airport	train	avg
Baseline	27.7%	34.7%	37.3%	39.9%	35.5%	42.2%	36.0%
Estimated ADBM	24.7%	31.2%	36.2%	35.8%	35.1%	37.5%	33.4%
Oracle ADBM	20.7%	24.2%	27.3%	24.9%	26.1%	26.5%	24.9%

Table 6.4: Word error rates for various mask types on the Aurora4 dataset. The estimated ASR-driven binary mask (ADBM) uses a lattice for hypotheses and a conservative mask for selecting a candidate mask at each frame. Word Error Rate for clean speech is 9.8%

set of candidates, the estimation task becomes a model selection problem. At each frame, the best frame mask must be selected from a set of candidate masks.

In our previously published pilot study [32], we presented a very simple method for selecting the best candidate. At each frame, the candidates were compared to a conservative mask and the candidate that best matched the conservative mask was selected. A conservative mask is a binary mask where decisions are only made at T-F units where we have high confidence in the decision. We began with a spectral subtraction based mask [6]. Using the first and last 20 frames of the utterance, we estimated the noise energy at each frequency band. From the noise estimate, we could estimate the speech energy at each T-F unit by subtracting the estimated noise energy from the mixed signal.

By comparing the noise and speech estimate at each T-F unit, we can create a binary mask. For the conservative mask, we use a large threshold. When the speech energy is at least 10 dB stronger than the noise, we assign a value of one. When the noise energy is at least 10 dB stronger than the speech, we assign a value of zero. Since we will not be using the mask directly, we do not need to assign a value to the remaining T-F units. Since the conservative mask only assigns values where it is highly confident, a majority of T-F units will not be assigned a value.

To estimate the ASR-driven binary mask, we compare the candidates at each frame to the conservative mask. The candidate that most closely matches is chosen. We do not consider the T-F units without an assigned value in this calculation. Results using this estimated mask are shown in Table 6.4. A modest decrease in WER is seen compared to the unenhanced baseline. While the WER improvements are not dramatic at this point, we have shown an estimated ASR-driven binary mask that does show some improvement. We have also outlined a simple framework for selecting a model from a list of candidates at each frame. The conservative mask may not be a good metric to compare against, but we can replace that mask with any baseline mask estimate. In the next section we will show how results can be improved using this simple framework and a better initial mask estimate as a comparison metric.

6.4 Baseline Mask Estimation Metrics

We have established a framework for estimating the ASR-driven binary mask. It requires a baseline mask estimate as a metric for comparison to the set of candidates at each frame. In the previous section we used a simple conservative mask for comparison. The main issue with the conservative mask is that it does not make any decisions for the majority of T-F units; it only makes decisions for T-F units that are relatively easy to correctly classify. In order to improve the mask estimation, we need a better initial mask estimation.

We introduce three different mask estimation methods. We will use each method as both a baseline for comparison and as a metric for selecting the best model at each frame. The first method is based on spectral subtraction and is described in Section 6.4.1. Section 6.4.2 discusses the parallel model combination based method of Kim and Hansen [54]. Both of these methods estimate the IBM, so there is a mismatch when using them to assist

Mask Type	car	babble	rest.	street	airport	train	avg
Baseline	27.7%	34.7%	37.3%	39.9%	35.5%	42.2%	36.0%
Conservative ADBM	24.7%	31.2%	36.2%	35.8%	35.1%	37.5%	33.4%
SS-Based Mask	24.6%	32.2%	35.3%	35.7%	35.2%	35.6%	33.1%
SS-Based ADBM	24.4%	30.8%	35.7%	35.4%	34.4%	37.2%	33.0%
Oracle ADBM	20.7%	24.2%	27.3%	24.9%	26.1%	26.5%	24.9%

Table 6.5: Word error rates for various mask types on the Aurora4 dataset. Baseline refers to unenhanced speech. Three estimated masks are used. Conservative ADBM estimates the ASR-driven binary mask with the conservative mask and SS-based ADBM uses an SS-based mask for estimation. The SS-based mask directly uses SS-based IBM estimate. Word Error Rate for clean speech is 9.8%

in the estimation of the ASR-driven binary mask. In Section 6.4.3, we use the prior two methods as features for direct estimation of the ASR-driven binary mask with an MLP.

6.4.1 Spectral Subtraction Based Mask

This binary mask estimation technique is based on spectral subtraction [6]. A simple estimate of the noise energy is obtained by averaging the energy in the silence regions of the utterance. For our experiments, we do not attempt to identify nonspeech regions; instead, we assume the first and last 20 frames in each utterance correspond to silence. Once we have a noise estimate, we can estimate the clean speech signal by subtracting the noise estimate from the mixture. Given the estimate for clean speech and noise, we can estimate the instantaneous SNR, the quantity required by the definition of the IBM. Once we have the instantaneous SNR, we can estimate the IBM by using a simple threshold.

The IBM is typically defined using a threshold of 0 dB, but a different threshold may match the IBM better when dealing with an estimate of the instantaneous SNR. We learn a frequency dependent threshold by maximizing mask accuracy on a training set. Both

masks are tested on the Aurora4 data set previously described in Chapters 4 and 5. Results are shown in Table 6.5.

The spectral subtraction (SS) based mask performs better than the baseline using both the 0 dB threshold and the learned threshold, but only results using the learned threshold are shown. In fact, performance is slightly better than the ASR-driven mask estimation method proposed in the previous section (*Conservative ADBM*). Since we are using an improved mask estimate, we can apply it to the ASR-driven mask estimation process. *SS-based ADBM* refers to using the SS-based mask estimate as our metric for identifying the best candidate at each frame. We use the SS-based mask with the trained threshold as it provides slightly better performance. It improves performance over the *Conservative-based ADBM* described in the previous section and is comparable to the SS-based mask.

Using the SS-based mask produces better performance than the previously proposed method for estimating the ASR-driven binary mask. However, incorporating the SS-based mask in the estimation process matches the performance of the SS-based baseline. Even though the new baseline mask estimation system improved over the previous method, we were able to utilize the new baseline to improve our system. We will see if that pattern continues as we use a better baseline mask estimation in the next section.

6.4.2 Posterior-Based Representative Mean

The posterior-based representative mean (PRM) binary mask estimation technique was originally proposed by Kim and Hansen in [52]. Motivated by ideas from Parallel Model Combination (PMC) [23], an estimate of the clean speech and noise-corrupted speech are compared to determine whether a T-F unit should be masked. We will briefly describe the method before reporting results.

A general clean speech model is first trained on the training data. While we will be making the masking decision in the linear spectral domain, the models are trained on cepstral features created by performing a DCT on the log spectral values. The clean speech is represented by a GMM; since we are modeling cepstral values, we can use Gaussians with diagonal covariances. A model of the noise-corrupted speech is also required. By estimating the underlying noise source with another model, the noise-corrupted model can be estimated by combining the clean speech and noise models. The noise is modeled by a single Gaussian in the same domain as the clean speech model. The noise estimate is obtained in the same way as with the SS-based mask; silence frames are assumed at the beginning and ending of each utterance and are used to build the noise model.

We adopt much of the notation from [54] for describing the method. The clean speech GMM model can be represented as

$$p(X) = \sum_{k=1}^{K} c_k \mathcal{N}(X; \mu_{X,k}, \Sigma_{X,k}) \tag{6.1}$$

where K is the number of components and c_k is the weight for each component. The noise model is represented by a single Gaussian pdf

$$p(N) = \mathcal{N}(\mu_N, \Sigma_N) \tag{6.2}$$

where N is the set of data used to estimate the noise.

The two models can be combined as

$$(\mu_{Y,k}, \Sigma_{Y,k}) = \mathcal{F}[(\mu_{X,k}, \Sigma_{X,k}), (\mu_N, \Sigma_N)] \tag{6.3}$$

where Y represents the noise-corrupted speech and $\mathcal{F}[\cdot]$ represents the function used to perform the model combination. Details describing exactly how the combination function is formulated can be found in [51]. The combination assumes the original linear spectral

data have a log-normal distribution and, therefore, a normal distribution in the log spectral domain.

The clean speech model is trained on a separate training set prior to testing. The noise model is trained separately for each utterance, so the combined noise-corrupted model is also created separately for each utterance. Given these models, the PRM estimate can be computed. The PRM estimate of the noise-corrupted speech $\hat{\mu}_Y(t)$ is defined as

$$\hat{\mu}_Y(t) = \sum_{k=1}^{K} p(k|Y(t))\mu_{Y,k} \tag{6.4}$$

where the posterior probability $p(k|Y(t))$ is given by

$$p(k|Y(t)) = \frac{c_k p(Y(t)|\mu_{Y,k}, \Sigma_{Y,k})}{\sum_{k=1}^{K} c_k p(Y(t)|\mu_{Y,k}, \Sigma_{Y,k})}. \tag{6.5}$$

A weighted sum of Gaussians from the noise-corrupted model is used to give the PRM estimate of the noise-corrupted speech. While the true noise-corrupted speech is observed, the PRM estimate also allows the weight for each Gaussian to be computed for the PRM estimate of clean speech $\hat{\mu}_X(t)$ in

$$\hat{\mu}_X(t) = \sum_{k=1}^{K} p(k|Y(t))\mu_{X,k}. \tag{6.6}$$

Now that the two PRM estimates have been computed, the mask can be estimated. The PRM estimates are first converted to the log spectral domain using the following transforms

$$\begin{aligned} \mu^{(log)} &= C^{-1}\mu \\ \Sigma^{(log)} &= C^{-1}\Sigma(C^{-1})^T \end{aligned} \tag{6.7}$$

Mask Type	car	babble	rest.	street	airport	train	avg
Baseline	27.7%	34.7%	37.3%	39.9%	35.5%	42.2%	36.0%
SS-Based Mask	24.6%	32.2%	35.3%	35.7%	35.2%	35.6%	33.1%
PRM-Based Mask	26.3%	32.4%	35.9%	39.3%	33.1%	43.0%	35.0%

Table 6.6: Word error rates for various mask types on the Aurora4 dataset. Baseline refers to unenhanced speech. The two estimated masks directly estimate the Ideal Binary Mask. Word Error Rate for clean speech is 9.8%

where C represents the discrete cosine transform. The mask is estimated by

$$M(f,t) = \begin{cases} 1 & \hat{\mu}_X^{(log)}(f,t) - \hat{\mu}_Y^{(log)}(f,t) > \theta_f \\ 0 & \text{otherwise} \end{cases} \tag{6.8}$$

where $M(f,t)$ is the mask value at frequency f and timeframe t. The threshold θ can be a single value, but we use a frequency dependent threshold to allow for a fair comparison with the SS-based mask estimation.

As with the SS-based mask estimation results in the previous section, we present results on the same Aurora4 corpus. The clean speech model used 512 Gaussians. The original study introducing the PRM-based mask estimation only used 128 [54], but we increased this number to compensate for the increased phonetic diversity of our corpus. Results are shown in Table 6.6. The *PRM-based mask* performs significantly worse than the *SS-based mask*, contrary to the results shown in [54]. There are several possible explanations. The PRM-based mask estimation may not perform as well on larger vocabulary datasets. The SS-based estimate may be a stronger baseline since we used a frequency dependent threshold. Previous work combined the PRM estimate with missing data techniques for recognition while we perform direct recognition from the masked speech without missing data compensation.

Even though the PRM-based mask performed worse than the SS-based mask, the errors may be somewhat uncorrelated. In the next section we will explore combining the two estimates to create an estimate that performs better than either method individually. We will also examine methods for directly estimating the ASR-driven binary mask.

6.4.3 Multilayer Perceptron Based Mask

We have introduced two previously published baseline methods for binary mask estimation and shown how they can be incorporated into the model selection process for ASR-driven binary mask estimation. One obvious deficiency in this approach is that the previous two techniques have a different goal. They seek to estimate the IBM, while we are estimating the ASR-driven binary mask. The two masks are different and using an estimate of one will not necessarily improve the estimate of the other. In this section we use the multilayer perceptron (MLP) to combine the multiple estimations. Our first set of experiments will still directly estimate the IBM, producing an even stronger baseline comparison method. Then we will use the MLP to estimate the ASR-driven binary mask. The MLP-based ASR-driven binary mask estimate should make a better metric for comparison during model selection as the targets are the same.

Ideal Binary Mask Target

We hypothesize that the SS-based and PRM-based mask estimation techniques make errors that are uncorrelated. For instance, the PRM-based estimate does not have an innate measure of SNR, while the SS-based method does estimate the SNR. By selecting a higher threshold for the SS-based method (the conservative mask), it produces estimates only for T-F units where the estimate is highly confident. We know the conservative mask can be very accurate in the T-F units it chooses to estimate. A simple method for combining the

Mask Type	car	babble	rest.	street	airport	train	avg
Baseline	27.7%	34.7%	37.3%	39.9%	35.5%	42.2%	36.0%
SS-Based Mask	24.6%	32.2%	35.3%	35.7%	35.2%	35.6%	33.1%
PRM-Based Mask	26.3%	32.4%	35.9%	39.3%	33.1%	43.0%	35.0%
PRM+Conservative	23.9%	30.0%	35.1%	35.5%	32.4%	36.0%	32.1%
PRM+SS MLP Mask	22.9%	29.5%	33.5%	35.0%	32.1%	35.7%	31.4%

Table 6.7: Word error rates for various mask types on the Aurora4 dataset. Baseline refers to unenhanced speech. The PRM+Conservative mask is a combination of the conservative and PRM masks. The PRM+SS MLP mask is an IBM estimate from an MLP using the SS and PRM estimates as features. Word Error Rate for clean speech is 9.8%

PRM-based mask and the SS-based mask is to use the SS estimate only in the regions of high confidence and the PRM estimate in all other regions.

As with previous experiments, we choose the threshold to be ± 10 dB. Results are shown in Table 6.7. The *PRM+Conservative Mask* significantly outperforms both the *PRM-Based Mask* and the *SS-Based Mask*. These results imply that the two previously described baseline estimation methods differ in the properties they model. Given that the combination of the two estimates in this heuristic manner proved beneficial, we further explore their combination through a more principled approach using an MLP.

One MLP is trained for each frequency band. The input layer contains two nodes, representing the PRM and SS estimates, and the hidden layer contains 8 nodes. Two outputs are used and provide the posterior probability of a T-F unit being masked or unmasked. The MLPs are trained on the same training set used for training the HMM acoustic models. The standard training set contains clean speech utterances, but the MLPs obviously needs data mixed with noise. We mix each training utterance with a either car, babble, factory, or white noise from the Noisex92 [95] database at a random SNR between 0 and 20 dB. Results using the MLP can also be seen in Table 6.7. The MLP-based combination produces a

Mask Type	car	babble	rest.	street	airport	train	avg
Baseline	27.7%	34.7%	37.3%	39.9%	35.5%	42.2%	36.0%
SS-Based ADBM	24.4%	30.8%	35.7%	35.4%	34.4%	37.2%	33.0%
MLP-Based ADBM	24.7%	30.1%	35.3%	35.8%	32.6%	36.1%	32.4%
Direct MLP ADBM	24.7%	29.8%	35.1%	34.9%	32.9%	35.5%	32.1%

Table 6.8: Word error rates for various mask types on the Aurora4 dataset. SS-Based ADBM uses the SS estimate to select the best model rom the set of hypotheses to estimate the ASR-driven binary mask (ADBM) and the MLP-Based ADBM uses the MLP trained to predict the ADBM for model selection. Direct MLP ADBM uses the output directly from the MLP as a mask estimate. Baseline refers to unenhanced speech.

result significantly better than any of the previously described baselines. We will use this strong baseline for comparisons against our estimation methods described in the following sections.

ASR-Driven Binary Mask Target

The MLPs trained to estimate the ASR-driven binary mask are trained in much the same way as those described for estimating the IBM. However, instead of using just the SS and PRM estimates, we introduce two additional features. We did not use these additional features for training IBM MLPs because their values would nott necessarily correlate well with the IBM target. The first additional feature is the value of the background prior. Recall that the background prior is compared to the subphonetic model to determine whether a T-F unit should be masked in the ASR-driven binary mask. It describes the relative energy in a particular frame compared to the entire utterance. Since the mask is defined based on this value, it follows that it should be used in its estimation. The final feature is the average value of every candidate mask in that frame. This feature provides the MLP with some information about the candidates.

Mask Type	car	babble	rest.	street	airport	train	avg
Baseline	27.7%	34.7%	37.3%	39.9%	35.5%	42.2%	36.0%
Ideal Binary Mask Target							
SS-Based Mask	24.6%	32.2%	35.3%	35.7%	35.2%	35.6%	33.1%
PRM-Based Mask	26.3%	32.4%	35.9%	39.3%	33.1%	43.0%	35.0%
PRM+Conservative	23.9%	30.0%	35.1%	35.5%	32.4%	36.0%	32.1%
PRM+SS MLP Mask	22.9%	29.5%	33.5%	35.0%	32.1%	35.7%	31.4%
ASR-Driven Binary Mask Target							
SS-Based ADBM	24.4%	30.8%	35.7%	35.4%	34.4%	37.2%	33.0%
MLP-Based ADBM	24.7%	30.1%	35.3%	35.8%	32.6%	36.1%	32.4%
Direct MLP ADBM	24.7%	29.8%	35.1%	34.9%	32.9%	35.5%	32.1%

Table 6.9: Word error rates for various mask types on the Aurora4 dataset. Compares all techniques using an IBM target and all techniques using an oracle ASR-Driven Binary Mask (ADBM) target. The best performing method, PRM+SS MLP Mask, significantly outperforms all other techniques in the average case.

Each MLP has 16 hidden units and two output units, corresponding to the probability of the mask being assigned zero and one respectively. The oracle ASR-driven binary mask is used as the target. As it directly estimates the ASR-driven binary mask, it should provide a better metric for model selection compared to estimates of the IBM. Results using the MLP-based mask estimate are shown in Table 6.8. *SS-Based ADBM* uses the SS estimate to select from a set of hypotheses and *MLP-Based ADBM* uses the output from the MLP just described. *Direct-MLP ADBM* shows that directly using the MLP output works just as well, if not better, as using it as a metric for model selection.

Table 6.9 shows the results from the IBM target based and ADBM based methods together. The best performing method is actually our *PRM+SS MLP Mask*. In fact, it significantly outperforms every other method presented. In order to further improve results, we need to better take advantage of the information the linguistic hypotheses provide us. So far, we have not used any temporal structure information in our estimation process.

In the next section we introduce an improved model selection algorithm that significantly outperforms the *PRM+SS MLP Mask* baseline and every other technique described in this chapter.

6.5 Improved Model Selection

At this point we have introduced an alternative masking criterion that forces the incorporation of the underlying linguistic information into the estimation of a binary mask. ASR hypotheses can be used to propose a small number of candidate masks at each frame. The problem of estimating the ASR-driven binary mask is reduced to a model selection problem where the best candidate mask must be selected at each frame. We proposed a simple framework for making this decision by comparing each candidate mask to a baseline mask estimate. While our estimated masks significantly outperform an unenhanced baseline, they do not outperform the baseline mask estimate used as a guide for model selection. In order to further improve results, we need leverage the linguistic information with more sophisticated techniques. In this section we propose a new framework for estimating the ASR-driven binary mask that does outperform each of the previously presented baseline systems.

6.5.1 Background

Our previously proposed framework suffered from two main deficiencies. The decision at each frame is made independently and when comparing the candidate models to a baseline mask estimate, each frequency channel is given equal weight. We propose a new framework that solves both of these deficiencies using a discriminatively trained sequence model. We will discuss several ways our problem can be transformed into this domain and then focus on a particular representation due to simplicity and computational efficiency.

Before we discuss various frameworks and whether they are appropriate for our problem, we review our prediction task. At each frame we have a set of subphonetic labels associated with the acoustic model of the HMM. In our experiments, these labels are triphone states. Each one of these labels has a simple prior model associated with it. From the prior model and information about the particular frame, we can produce a frame-level mask based on the definition of the ASR-driven binary mask. Our ultimate goal is the final binary mask estimate. We could find the best subphonetic model at each frame which would in turn define a binary mask or we could ignore the subphonetic models and directly find the best mask at each frame.

We could directly estimate the binary mask by viewing the estimation process as an image segmentation problem. The spectral representation can be represented as an image and a graph cut algorithm [55] can be employed to segment the T-F units into a binary mask. There has been some work in applying this approach to mask estimation [103], but it does have its disadvantages. A spectral representation of speech is inherently different than a visual image as the dimensions of the images are not necessarily comparable. On the x-axis we have time and on the y-axis we have frequency. A simple graph structure would simply have edges between adjacent T-F units, but this adjacency may need to be handled differently between frequency and time. Also, due to the effects of harmonics, it may be desirable to have edges between nonadjacent T-F units on the frequency domain. As the number of edges increases, so does the difficulty of computing the graph-cut.

As we would like to avoid the issues inherent in a graph structure, we focus on algorithms that use a linear chain structure. One popular model is the linear chain conditional random field (CRF) [56]. The CRF is discriminatively trained to maximize the conditional

likelihood of an entire sequence of labels. Originally proposed for tasks in natural language processing, it has been applied to both phone recognition [70] and word recognition [106].

If we want to use a linear chain sequence model for our task, the first question is what is the label space. We could either predict the subphonetic label, complete frame mask, or a combination of the two. If we choose to predict the subphonetic label, our task becomes similar to the phone recognition task. The main difference is that in the phone recognition tasks previously mentioned, the size of the label space is typically less than 100. We have over 20,000 subphonetic models, so our label space would be much larger. Also, there may be significant differences in the properties of a label depending on the associated mask. In that case, we would want our label space to be the cross product of the subphonetic models and the frame masks, resulting in $20,000$ x 2^{64} total labels. Obviously a label space of this size would be unmanageable due to computational reasons and data sparsity concerns.

6.5.2 Model Description

We propose a discriminative linear chain sequence model trained using the structured perception algorithm [11]. In general, this requires learning a weight associated with each of a large set of feature functions. We still have the problem of an immense label space. Learning the weights associated with the combination of every feature function and every possible label would be impossible. An approximation must be made and we accomplish that by the way we associate the feature functions with the labels. Instead of learning a weight for a feature function associated with every possible label, we associate a feature function to a feature of the label. For instance, we could have a weight for a feature function when the label is one for the 0th frequency channel. We could have a weight for a feature

Triphone Context	Freq. Dependent	Feature Type
Observation Feature Functions		
Left Context	no	bias
Right Context	no	bias
Left Context	yes	posterior
Right Context	yes	posterior
Center Context	no	bias
Center Context	yes	bias
Center Context	yes	posterior
none	yes	bias
none	yes	posterior
Transition Feature Functions		
Center Context	no	bias
Center Context	yes	bias
none	yes	bias

Table 6.10: List of feature functions used in the sequence model. Triphone context refers to functions related to a portion of the triphone. Frequency dependence describes whether the function is general or has a version for each of the 64 frequency bins. Feature type refers to whether the value of the feature is a bias or an MLP posterior.

function when the left context of the triphone is /ah/. In Table 6.10 we have a list of how we create all of the feature functions.

The bias features are straightforward and always evaluate to a value of one when fired. The posterior probability features are associated with the MLPs described in Section 6.4.3. Each frequency has its own MLP which produces two outputs, the posterior probability of a particular T-F unit being labeled 0 and 1 respectively. Since we have two features for every frequency channel, we have 128 total posteriors for every frame.

Another way of thinking about the feature functions is that we have factored the label space. We have feature functions associated with the left, right, and center context of the triphone and each individual frequency channel. Any given feature function will fire for a large number of labels, but a unique set of feature functions will be nonzero for each

particular label. This allows us to maintain a manageable number of weights that can be learned, but still provide some amount of discrimination between labels.

Our new model will produce a score associated with any label sequence. We can now estimate the ASR-driven binary mask by finding the label sequence that maximizes

$$\underset{y}{\operatorname{argmax}} \sum_{i} \sum_{k} \alpha_k f_k(y_i, y_{i-1}, x_i) \tag{6.9}$$

where i is the frame index, k is the feature function index and α_k is the weight associated with the feature function $f_k(\cdot)$. The x_i vector contains the posteriors predicted by the MLP at that frame. Note that while the label sequence y can usually be efficiently found using the viterbi sequence, the number of labels is still far too large in our case.

Due to the way we have formulated the problem, however, we can make a simple approximation. The first pass of the ASR system has produced a set of candidate labels at each time frame. Since we only plan on considering the candidate models, we only have a small set of possible labels at each frame. By considering only the candidate set instead of the entire label space, solving the above equation becomes tractable.

We train our model using the averaged structured perceptron algorithm [11] using the same training data described in Section 6.4.3 for training the MLPs. The structured perceptron algorithm requires two pieces of information, the current maximum scoring label sequence and the true label sequence. The true sequence is obtained for force-aligning the clean speech to obtain a triphone state label at each frame. The mask associated with each triphone state at a particular frame is then computed to obtain the full label. The current maximum label sequence is found simply by evaluating Equation 6.5.2 for the set of possible candidates.

We do make one small change to the perceptron update. Typically any feature function associated with an incorrect label is updated. However, in our case, we are only concerned

Mask Type	car	babble	rest.	street	airport	train	avg
Baseline	27.7%	34.7%	37.3%	39.9%	35.5%	42.2%	36.0%
Direct MLP ADBM	24.7%	29.8%	35.1%	34.9%	32.9%	35.5%	32.1%
PRM+SS MLP Mask	22.9%	29.5%	33.5%	35.0%	32.1%	35.7%	31.4%
SM-Based ADBM	23.8%	28.5%	32.1%	34.0%	31.1%	35.0%	30.7%

Table 6.11: Word error rates for various mask types on the Aurora4 dataset. Comparison of the proposed sequence model (SM) based mask estimation and the previous best ASR-driven binary mask estimator and the best baseline IBM estimation.

that the frame mask portion of the label is correct. If the correct mask is hypothesized, but uses a different triphone state sequence, it makes no difference to our final output. Due to this fact, we make no updates to feature functions associated with a particular frame if the mask for that frame is correct. This eliminates situations where the weights are continually updated even when the mask is correct.

6.5.3 Results

Our improved model selection algorithm is evaluated on the same Aurora4 corpus previously used and compared to the baseline systems outlined in Section 6.4. Results are shown in Table 6.11. *SM-Based ADBM* is the sequence model based system just described. *Direct MLP ADBM* is the MLP trained with the ASR-driven binary mask as the target and *PRM+SS MLP Mask* is the MLP trained with the IBM as the target. In the average case, our proposed model significantly outperforms the best baseline method we have explored.

The proposed model provides a large improvement over directly estimating the ASR-driven binary mask using the MLP-based system. The local T-F unit mask estimate provided to the *SM-based ADBM* resulted in significantly worse performance than the baseline

system. However, by utilizing the sequence information implicitly provided by the linguistic models and the context-dependent frequency weighting, the proposed system could overcome this poor starting estimate to outperform the *PRM+SS MLP Mask* system. This result is promising; perhaps if the proposed model had access to a better list of candidate hypotheses or a better initial mask estimation, it could see further improvement.

While our results compare well to the comparisons we have tested, they are not state of the art. Previously published results on Aurora4 have been as low as 14% word error [82, 93]. Comparing the reported results for different ASR systems can be difficult; the difference in results can be due to a large number of factors. For instance, the 14% word error results use multi-condition trained acoustic models. If the acoustic models are trained on clean speech, the results increase to approximately 21% word error [93, 85]. The most similar published system to our own is the previously discussed Srinivasan et al. [92] system, but many differences still exist; we use variance normalized features, a different mask estimation approach, and do not perform reconstruction. Our system is significantly better, but the improvement cannot be attributed to just one factor.

6.6 Conclusions

We have addressed the issue of incorporating estimated binary masks in ASR. By using non-zero values for the masked units, the estimated units are able to perform better than baseline unenhanced recognition. We proposed a simple method of estimating the ASR-driven binary mask by selecting the model from a list of candidates based on the model most closely matching a baseline estimate. However, this simple model selection process

did not perform as well as our best performing baseline method. We also proposed a discriminatively trained sequence model for model selection that did outperform all baseline systems.

By utilizing the implicit sequence information encoded in the candidate models, we were able to significantly improve ASR performance. Performance would likely further increase from better initial hypotheses or from improved baseline mask estimation. Our reliance on the ASR system to generate hypotheses does have limitations; interference from another speaker would be very difficult to handle and the system is obviously not real-time. In the next chapter, we will investigate the effect of a reduced number of candidate models per frame. We also investigate whether recognition using the *SM-Based ADBM* will improve the quality of ASR lattices and provide better candidate models for a second pass mask estimation.

CHAPTER 7: CLOSING THE LOOP

In Chapter 5 we introduced the ASR-driven binary mask. By tying the masking decision to the acoustic model of an ASR system, the estimated mask is forced to utilize linguistic information. Chapter 6 presented a method for estimating the ASR-driven binary mask that outperformed all presented baseline systems. We have largely only considered obtaining candidate models for mask estimation from a word lattice output from an ASR system recognizing unenhanced speech. In this chapter we explore two variations on the set of candidate hypotheses. We examine the effect of reducing the number of candidate models, either by restricting the number of models used from the lattice or obtaining an N-best list from the recognizer, on mask estimation performance. We also outline how the ASR-driven binary mask could be estimated through an iterative approach.

7.1 Reduced Candidate Experiments

For our estimation experiments, we have chosen to use a word lattice to generate candidate models at each time frame. The lattices we generate contain not only word sequence information, but time alignment and pronunciation information. Even for the same word sequence there may be multiple hypotheses for the time alignments of the words. We consider this type of lattice to be an upper bound on generating candidate hypotheses.

Another method for generating hypotheses is through an N-best list. The results of the N-best list are different from the lattice since the N-best contains no alignment information. Given each utterance in an N-best list, we force align the word sequence to the data. Since

Mask Type	car	babble	rest.	street	airport	train	avg
Baseline	27.7%	34.7%	37.3%	39.9%	35.5%	42.2%	36.0%
Estimated Masks							
1-Best Estimate	25.2%	32.5%	34.3%	35.4%	33.8%	36.6%	33.0%
100-Best Estimate	23.8%	29.0%	33.0%	34.8%	31.6%	35.6%	31.3%
Lattice Estimate	23.8%	28.5%	32.1%	34.0%	31.1%	35.0%	30.7%
Oracle Masks							
100-Best Oracle	22.8%	28.8%	31.8%	31.7%	29.6%	31.2%	29.3%
Lattice Oracle	20.7%	24.2%	27.3%	24.9%	26.1%	26.5%	24.9%

Table 7.1: Word error rates for various mask types on the Aurora4 dataset. Compares the results of estimating the ASR-driven binary mask when using candidate models generated from a 1-best list, 100-best list, and word lattice. Comparisons against oracle masks are also presented.

the forced alignment always produces the same result given the word sequence, we can never obtain the same diversity of hypotheses as with the lattice. The lattice produces an average of 8.7 candidates per frame while a 100-best list produces an average of 1.7 candidates per frame.

One question is how the reduction in candidate hypotheses affects performance. We answer this question by considering three ways of generating hypotheses, the lattice, 100-best list, and 1-best list. The 1-best list is simply the standard output of a recognizer and produces exactly one hypothesis per frame. We generate each of the masks using the best performing system from the previous chapter, the sequence model based mask estimation system. Results on the Aurora4 dataset are shown in Table 7.1.

While using the lattice produces the best result, the highly constrained hypotheses produced by the 100-best lattice do not reduce accuracy as much as expected. More surprising is the improvement seen by simply using the 1-best output. Since the 1-best output produces only one candidate model per frame, no feature generation is required; we do not

need to evaluate the utterance using the SS or PRM estimation. The sequence model is also unnecessary because we have no model selection problem. Simply using the incorrect 1-best output, we can reduce WER by an absolute of 3%.

Recall that the definition of the ASR-driven binary mask uses no information about the actual energy in a T-F unit. Each unit is masked based on the expectation of energy in a T-F unit given a particular state model. Frames with less energy are more likely to be masked, but it does not influence the decision at the T-F unit level. Simply by using the output of the ASR recognizer, we can obtain a mask that works as well as a SS-based mask for ASR. In our system, a simple estimate of the linguistic content works as well as a simple estimate of the interfering signal. We believe the *100-Best Estimate* and *Lattice-Best Estimate* demonstrate the benefit of combining these two sources of information.

The gap between the number of candidates produced between the lattice and 100-best list is large. In order to investigate the use of differing numbers of candidate models between those two results, we present experiments where we artificially restrict the number of candidates considered from the lattice. The candidate models at each frame are ordered in terms of the acoustic likelihood produced by the ASR system. We restrict the number of candidates by either setting a maximum number per frame or by only considering candidates within a certain beam width of the maximum likelihood in that frame.

In our first set of reduced experiments, we set the maximum number of candidates per frame to be between 10 and 25 with increments of 5. Results showing WER versus the average number of candidates per frame are shown in Figure 7.1. Instead of showing the maximum number of candidates in the x-axis, we show the average. Notice there is also a plot at 1 and 1.7 candidates per frame. Those points are from the 1-best and 100-best results respectively and are displayed for completeness. Limiting the total maximum

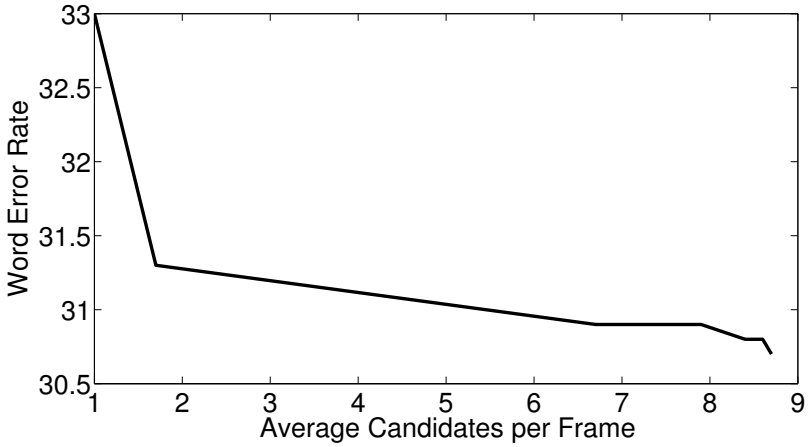


Figure 7.1: Word error rates vs. Average number of candidates per frame for the Aurora4 dataset. The number of candidates is varied by setting a maximum number per frame.

number of candidates to 10 reduces the average number of candidates by less than three. It also appears to have no effect on the WER.

For our second set of experiments, we set a beam width instead of a maximum number of candidates. The beam width for the log likelihood of each state is set between 10 and 25 with increments of 5. Results are in Figure 7.2 and follow the same format as those from Figure 7.1. The beam width gives a larger variation in average candidates per frame. We also see some variation in WER. It is possible that limiting the candidates by acoustic likelihood causes some issues due to the inherent mismatch between the clean models and noisy data. We had hypothesized that limiting the candidates based on likelihood might actually improve performance. Obviously this hypothesis was incorrect for the unenhanced speech, but we will also investigate this hypothesis for the improved second pass lattices since the models should better match the masked speech.

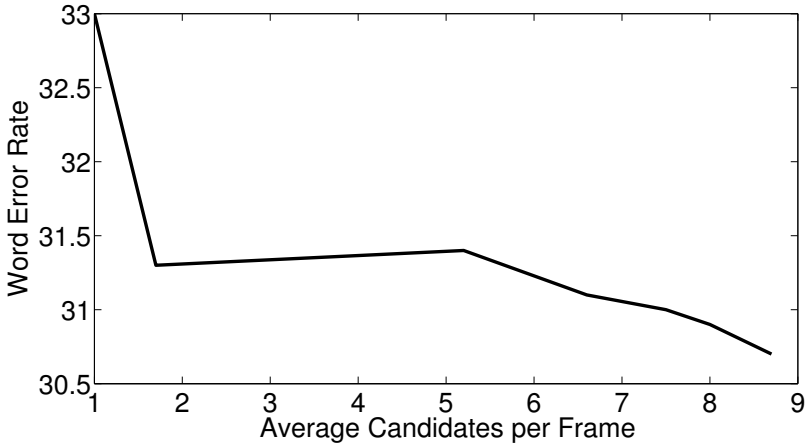


Figure 7.2: Word error rates vs. Average number of candidates per frame for the Aurora4 dataset. The number of candidates is varied by setting a maximum local beam width at state.

7.2 Iterative Mask Estimation

One of the strengths of the ASR-driven binary mask is the potential for iteratively improving the mask. The motivation for this idea comes from the early work of Ellis [17]. He proposed an idea that he termed the prediction-driven approach to computational auditory scene analysis. His ideas dealt with a broader scope than just robust ASR; multiple sources would all be identified, separated, and described. Predictions at one level would guide the predictions at another level and through an iterative process an explanation of the signal would finally be formed.

In [18], he outlined how his approach would directly relate to robust ASR. Essentially the system would consist of two main modules, a speech and nonspeech element recognizer. In the initial step, both modules would generate hypotheses to fully explain the observed data. Then both modules would attempt to incorporate the hypotheses of the other module into their improved explanation. This iterative process would continue until the best joint

hypothesis is found. At a high level, the proposed idea is simple, but each module could be very complex. The speech module may contain both low-level and high-level information. Low-level information could include fundamental frequency and formant track estimates or speech landmark detectors [34]. High-level information could be driven by the ASR system to provide word estimates or more general information about speaking rate and speaker identity. Integrating the hypotheses of the two modules or even different pieces of the same module is not straightforward.

In the original work, only high level implementation details were outlined and experimental evaluations were missing. Since the presentation of these ideas, little work has been published attempting to implement them. Srinivasan and Wang [90] proposed a phonemic restoration system that incorporates some of these ideas. They assume a task where a random phoneme in an utterance has been entirely masked by a high energy, broad spectrum noise source. A mask is estimated and recognition is performed using a missing data ASR system. Due to the constrained vocabulary and language model, the missing data system can still recognize the utterance with high accuracy. Given the ASR hypothesis, the masked phoneme can be reconstructed with a representative template matching the hypothesized phoneme. The mask estimation can be viewed as making a hypothesis about the noise source. The ASR component then integrates this information to develop a hypothesis about the speech. Based on the two hypotheses the missing speech is reconstructed. No iterative process exists, but the system is at least partially motivated by [18] and incorporates some of the basic ideas.

Barker et al. [1] also proposed a system using similar ideas. Instead of taking an iterative approach, a standard ASR system is modified to jointly predict the word sequence and a binary mask to separate two sources. Though the process is not iterative, it is still very

similar to the ideas proposed in [18]. The system finds an explanation that best explains both the speech and the interfering noise. Computational cost was the major drawback to the system as decoding over every possible binary labeling of the units in the spectral domain was expensive. A simplification was made where groups of T-F units were first found such that the labeling was only performed for entire groups as opposed to individual T-F units. This improved the efficiency of the system, but at the cost of potential accuracy due to errors in the initial grouping. The final recognition could potentially guide a second attempt at grouping T-F units, but this was not explored.

Our proposed system can also be viewed as a system that uses similar ideas. An initial pass through the recognition system hypothesizes speech explanations for the data. Low level mask estimation approaches, such as spectral subtraction (Section 6.4.1) and posterior-based representative mean estimation (Section 6.4.2), hypothesize noise explanations by identifying T-F units largely corrupted by noise. We combine these hypotheses to estimate the ASR-driven binary mask and recognition is performed on the partially masked signal. By using the ASR output as new, presumably improved hypotheses, we can use an iterative mask estimation approach. Experimental results using the iterative approach are shown in the next section.

7.3 Second Pass Mask Estimation Results

We generate the results for the second pass in the same manner as the first pass; candidate models are generated from the lattice, features are generated, and the MLP posteriors are computed. Results comparing the first pass and second pass estimation can be seen in Table 7.2. The second pass does not produce a significant improvement in terms of

System	car	babble	rest.	street	airport	train	avg
Baseline	27.7%	34.7%	37.3%	39.9%	35.5%	42.2%	36.0%
First Pass Lattice	23.8%	28.5%	32.1%	34.0%	31.1%	35.0%	30.7%
Second Pass Lattice	23.6%	28.1%	32.9%	33.1%	31.2%	34.4%	30.5%

Table 7.2: Word error rates on the Aurora4 dataset using the first and second pass lattices for candidate model hypothesis generation

System	car	babble	rest.	street	airport	train	avg
First Pass Lattice	9.8%	13.5%	16.2%	18.8%	15.0%	19.6%	15.5%
Second Pass Lattice	8.0%	11.5%	15.3%	15.1%	13.0%	16.0%	13.1%

Table 7.3: Oracle lattice error rates on the Aurora4 dataset using the first and second pass lattices.

WER. It is possible the improvements from the first pass are not great enough to see further improvements in the second pass, however, we can also compare the results of the two iterations in other ways.

One measure of the quality of hypotheses produced by the ASR lattice is the oracle lattice error rate (LER). As the LER decreases, the likelihood of the true model being present in the set of candidate hypotheses should increase. LER results for the two lattices are shown in Table 7.3. By enhancing the speech signal with the first pass mask estimation, the oracle error rate for the lattices has significantly decreased.

Another possible way of analyzing the lattice is to examine the effects of reducing the number of candidates as in Section 7.1. We again reduce the number of candidates by setting a local beam width. Results comparing the two lattices can be seen in Figure 7.3. The second pass lattice is consistently better than the first pass lattice and effect of reducing the number of candidates based on acoustic model scores is smaller. Based on this result,

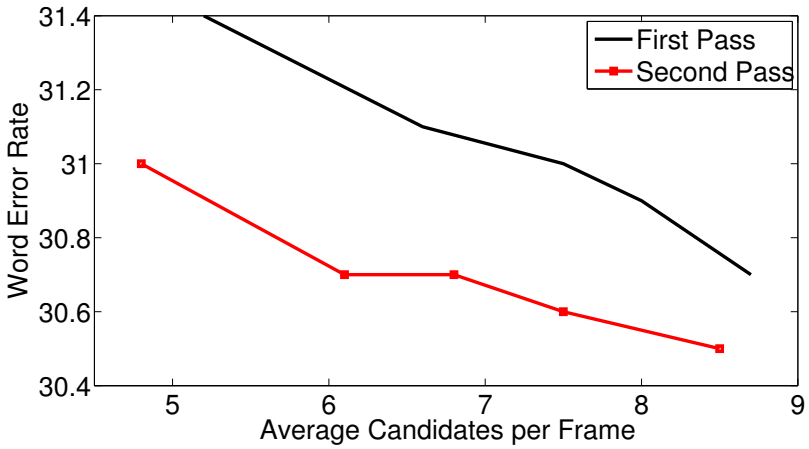


Figure 7.3: Word error rates vs. Average number of candidates per frame for the Aurora4 dataset. The number of candidates is varied by setting a maximum local beam width at state. Comparison between results of the first lattice and the second generated lattice.

System	car	babble	rest.	street	airport	train	avg
First Pass Lattice	96.0%	95.4%	94.9%	94.4%	95.4%	94.4%	95.1%
Second Pass Lattice	96.2%	95.6%	95.1%	94.7%	95.6%	94.8%	95.3%

Table 7.4: Mask accuracy results for first and second pass lattices on Aurora4. The oracle ASR-driven binary mask is treated as the true mask for the accuracy calculation

the acoustic scores at the state level may provide a better source of information in later iterations.

Since our focus has been on robust ASR, we have not reported results on metrics involving the mask itself. However, to further illustrate the improvement seen by estimating the mask from the improved mask, we do examine mask accuracy. As our goal is the ASR-driven binary mask, we will consider the oracle ASR-driven binary mask to be the correct mask. Results for the estimated masks based on the first and second pass lattices are shown in Table 7.4. The mask estimated from the improved lattice is consistently better in terms

of mask accuracy. Even though this improvement did not translate to improved ASR results, it shows the second iteration did produce some manner of improvement in the mask estimation.

7.4 Conclusions

We examined the effects of reducing the number of candidates at each frame on the mask estimation process. Even reducing the number of candidates to one produced a significant improvement over an unenhanced baseline; the noisy hypotheses generated by the unenhanced baseline still offer some information for mask estimation. Using candidates generated by a 100-best list further reduces error rate and the using the lattice produces our best result. Reducing the number of candidates by limiting the maximum number of candidates per frame had little effect on performance, but using a local beam on the acoustic likelihoods produced a stronger effect; the local acoustic likelihoods may not be a good metric when using unenhanced noisy speech.

Since our mask estimation algorithm depends upon the ASR system to produce candidate hypotheses, it is possible an iterative estimation process can be used. Our first pass mask estimate enhances the speech and a new lattice is generated. Based on the oracle lattice error rate of the two lattices, the second pass lattice contains more correct hypotheses. However, the improvements to WER when estimating a mask from the second pass lattice were not statistically significant. We also examined the improvements to overall mask accuracy and found the second pass lattice gave consistent improvements over the first pass lattice, producing an average 4% relative decrease in error.

One potential issue with iterative mask estimation is the process simply reinforcing the mistakes of the previous iteration. However, it appears that the second iteration did produce

better acoustic likelihoods for the state models. Reducing the number of candidates using a local beam associated to the likelihood was consistently better for the second pass. Iterative estimation is costly and future work will need to consider whether the improved results are worth the cost.

CHAPTER 8: CONCLUSIONS

8.1 Contributions

We have investigated methods for incorporating the binary mask in ASR. Previous work has shown that the binary mask cannot be used directly [12], but the reasons were not thoroughly investigated. We identified both a cause for the poor performance of binary masked speech in ASR and a possible solution. While the introduction of noise typically decreases the variance in the cepstral features, masking the noise-dominant regions increases the variance compared to features calculated from clean speech data. By normalizing the variance of the acoustic features, the IBM can be used directly without the use of methods from missing data ASR [13, 83].

Typical binary mask estimation methods utilize low-level features features and ignore the higher level linguistic information in the signal. We proposed an alternative masking criterion to create an ASR-driven binary mask. The masking decision was tied directly to the alignment produced between the true word sequence and the acoustic models of the ASR system. By obtaining n-best lists or word lattices as output from the ASR system, multiple masking hypotheses can be created for each frame, transforming the mask estimation problem to a frame level model selection problem.

We proposed a method for estimating the ASR-driven binary mask using a discriminatively trained linear sequence model. Masking decisions are made at the frame level and model temporal constraints both in the mask and the subphonetic models associated with

the masks at each frame. Posterior estimates of binary label at each time-frequency unit are utilized as features and other baseline methods for estimating the binary mask could also be incorporated as features. Our proposed estimation system outperforms several baseline methods by utilizing higher level linguistic information about the signal. The estimation method can be extended to work as an iterative estimation method. While our iterative experiments did not significantly improve ASR performance, the accuracy of the mask was improved.

8.2 Future Work

Many interesting questions remain associated with the work explored in this thesis. We outline several potential avenues for future work. In Section 3.3, we briefly discussed two spectro-temporal representations for speech, the spectrogram and the cochleagram. Another common representation is the mel-scale spectrogram, a transformation of the frequency axis of the standard spectrogram from a linear scale to the mel-scale. An investigation into the relative strengths of these representations for robust ASR needs to be made. We have found better recognition performance for ideal binary masks in the spectrogram domain than in the cochleagram domain, though the mask may be easier to estimate in the cochleagram domain. It is unknown if these observations hold for estimated masks. Whether the effect of making errors is dependent on the domain is also unknown.

Our proposed method for binary mask estimation estimates the ASR-driven binary mask, but the IBM has potential for better absolute performance. The key requirement for our model is an association between frame level binary masks and the subphonetic acoustic models in the ASR system. The definition for the ASR-driven binary mask provides this correspondence in a simple, straightforward manner. Future work could investigate

methods for training model-dependent binary mask estimation algorithms. If highly accurate mask estimation algorithms could be built for each subphonetic model, our algorithm could be applied directly to the estimation of the IBM.

A disconnect exists between standard methods of training binary mask estimation algorithms and robust ASR performance. Our model optimizes mask accuracy, but we have seen results where a more accurate mask performed significantly worse in terms of ASR performance. Measures such as HIT-FA [43] and IBMR [46] have been proposed to better match relationships between the estimated mask and performance on speech intelligibility for humans. However, the relationship between these measures and ASR results is unknown. A thorough study of the correspondence between estimated mask measures and ASR errors could potentially improve training criterion, and ultimately, mask estimation.

For our estimated masks we used a nonzero value for the masked T-F units due to the issues of calculating statistics for variance normalization. Using a nonzero value decreased the potential performance of our estimated masks in terms of word error rate. While variance normalization is clearly helpful in the case of both oracle masked speech and unenhanced speech, it introduces issues for estimated masks. Future work could likely achieve better performance using the same estimated masks presented in this thesis by improving methods for incorporating those estimated masks.

Our method of estimation also motivates further study in spectral reconstruction techniques. One of the difficulties in reconstructing the spectrum for missing data ASR is the general speech models used. If the speech models were more specific, it follows the reconstruction could be improved [72]. Since our mask estimation method not only estimates the mask, but also hypothesizes a specific subphonetic model at each frame, a more specific speech prior could be applied for reconstruction. How beneficial these specific models

would be, especially in the presence of both mask errors and model estimation errors, remains to be seen.

BIBLIOGRAPHY

[1] J. Barker, M. Cooke, and D. P. W. Ellis. Decoding speech in the presence of other sources. *Speech Communication*, 45:5–25, 2005. 57, 58, 110

[2] J. Barker, M. Cooke, and D. P.W. Ellis. The RESPITE-CASA-Toolkit Project. Available: http://staffwww.dcs.shef.ac.uk/people/J.Barker/ctk.html, 2002. 38

[3] J. Barker, L. Josifovski, M. Cooke, and P. Green. Soft decisions in missing data techniques for robust automatic speech recognition. In *Proceedings of International Conference on Spoken Language*, pages 373–376, Beijing, China, 2000. 38, 39

[4] L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annuls of Mathematical Statistics*, 41(1):164–171, 1970. 11

[5] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003. 13

[6] S. F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27:113–120, 1979. 1, 26, 56, 86, 88

[7] A. S. Bregman. *Auditory Scene Analysis*. MIT Press, Cambridge, Mass, 1994. 18

[8] G. J. Brown and M. Cooke. Computational auditory scene analysis. *Computer Speech and Language*, 8:297–336, 1994. 19

[9] D. Brungart, P. S. Chang, B. D. Simpson, and D. L. Wang. Isolating the energetic component of speech-on-speech masking with ideal time-frequency segregation. *Journal of Acoustical Society of America*, 120:4007–4018, 2006. 28

[10] C. Cherry and C. Quirk. Discriminative, syntactic language modeling through latent svms. In *Proceeding of AMTA*, 2008. 13

[11] M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*, 2002. 99, 101

[12] M. Cooke, P. Green, and M. Crawford. Handling missing data in speech recognition. In *Proceedings of ICSLP*, 1994. 1, 2, 28, 33, 116

[13] M. Cooke, P. Green, L. Josifovski, and A. Vizinho. Robust automatic speech recognition with missing and unreliable acoustic data. *Speech Communication*, 34:267–285, 2001. 1, 2, 29, 37, 38, 39, 57, 65, 116

[14] M. Cooke, A. Morris, and P. Green. Recognising occluded speech. In *Proceedings of ESCA Workshop on the Auditory Basis of Speech Perception*, pages 297–300, 1996. 33, 35, 36, 46, 49

[15] S. B. Davis and P. Mermelstein. Comparison of parametric representation for monosyllable word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28:357–366, 1980. 6, 30

[16] L. Deng, J. Droppo, and A. Acero. Dynamic compensation of HMM variances using the feature enhancement uncertainty computed from a parametric model of speech distortion. *IEEE Transactions on Speech and Audio Processing*, 13:412–421, 2005. 1

[17] D. P. W. Ellis. *Prediction-driven computational auditory scene analysis*. PhD thesis, MIT, 1996. 109

[18] D. P. W. Ellis. Using knowledge to organize sound: The prediction-driven approach to computational auditory scene analysis and its application to speech/nonspeech mixtures. *Speech Communication*, 27:281–298, 1999. 109, 110, 111

[19] D. P.W. Ellis, J. A. Bilmes, E. Fosler-Lussier, H. Hermansky, D. Johnson, B. Kingsbury, and N. Morgan. The SPRACHcore software package. Available: http://www.icsi.berkeley.edu/~dpwe/projects/sprach/sprachcore.html, 2010. 35, 49

[20] M. Fleischman and D. Roy. Grounded language modeling for automatic speech recognition of sports video. In *in Proceedings of HLT/NAACL*, 2008. 13

[21] S. Furui. Speaker-independent isolated word recognition using dynamic features of speech spectrum. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(1):52–59, 1986. 8

[22] M. J. F. Gales and S. Young. The Application of Hidden Markov Models in Speech Recognition. *Foundation and Trends in Signal Processing*, 1:195–304, 2007. 4

[23] M. J. F. Gales and S. J. Young. Robust Continuous Speech Recognition using Parallel Model Combination. *IEEE Transactions on Speech and Audio Processing*, 4:352–359, 1996. 1, 17, 89

[24] J. L. Gauvain and C. H. Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Transactions on Speech and Audio Processing*, 2:291–298, 1994. 17

[25] J.F. Gemmeke and B. Cranen. Noise reduction through compressed sensing. In *Proc. Interspeech*, 2008. 31, 45

[26] B. R. Glasberg and B. C. J. Moore. Derivation of auditory filter shapes from notched-noise data. *Hearing Research*, 47:103–138, 1990. 23

[27] Y. Gong. Speech Recognition in Noisy Environments: A Survey. *Speech Communication*, 16:261–291, 1995. 1, 16, 17

[28] J. Goodman. A bit of progress in language modeling. *Computer Speech and Language*, pages 403–434, October 2001. 13

[29] H. Gustafsson, S. E. Nordholm, and I. Claesson. Spectral Subtraction Using Reduced Delay Convolution and Adaptive Averaging. *IEEE Transactions on Speech and Audio Processing*, 9:799–807, 2001. 18

[30] X. Haitian, P. Dalsgaard, T. Zheng-Hua, and B. Lindberg. Robust Speech Recognition by Model Adaptation and Normalization Using Pre-Observed Noise. *IEEE Transactions on Audio, Speech, and Language Processing*, 15:2431–2443, 2007. 16

[31] W. Hartmann and E. Fosler-Lussier. Investigations into the incorporation of the ideal binary mask in ASR. In *Proceedings of IEEE ICASSP*, pages 4804–4807, Prague, Czech Republic, 2011. 33, 49, 54, 65

[32] W. Hartmann and E. Fosler-Lussier. ASR-driven top-down binary mask estimation using spectral priors. In *Proceedings of IEEE ICASSP*, pages 4685–4688, Kyoto, Japan, 2012. 54, 75, 76, 86

[33] W. Hartmann, A. Narayanan, E. Fosler-Lussier, and D. L. Wang. A direct masking approach to robust ASR. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(10):1993–2005, 2013. 33, 38

[34] M. Hasegawa-Johnson, J. Baker, S. Greeberg, K. Kirchhoff, J. Muller, K. Sonmez, S. Borys, K. Chen, A. Juneja, K. Livescu, S. Mohan, E. Coogan, and T. Wang. Landmark-based speech recognition: Report of the 2004 johns hopkins summer workshop. Technical report, Johns Hopkins Center for Speech and Language Processing, 2005. 110

[35] R. C. Hendriks, R. Heusdens, and J. Jensen. MMSE based noise PSD tracking with low complexity. In *Proceedings of IEEE ICASSP*, pages 4266–4269, 2010. 56

[36] H. Hermansky, Brian A. Hanson, and H. Wakita. Peceptually Based Linear Predictive Analysis of Speech. In *Proceedings of ICASSP*, volume 10, pages 509–512, 1985. 17

[37] H. Hermansky, N. Morgan, and H.-G. Hirsch. Recognition of Speech in Additive and Convolutional Noise Based on RASTA Spectral Processing. In *Proceedings of ICASSP*, volume 10, pages 509–512, 1993. 17

[38] J. N. Holmes and W. Holmes. *Speech Synthesis and Recognition*. CRC Press, 2001. 7, 14

[39] G. Hu. *Monaural speech organization and segregation*. PhD thesis, The Ohio State University, 2006. 21

[40] G. Hu and D. L. Wang. Monaural speech segregation based on pitch tracking and amplitude modulation. *IEEE Transactions on Neural Networks*, 15(5):1135–1150, 2004. 18, 21, 57

[41] G. Hu and D. L. Wang. Segregation of unvoiced speech from nonspeech interference. *The Journal of the Acoustical Society of America*, 124:1306–1319, 2008. 20, 21

[42] K. Hu and D. L. Wang. Incorporating spectral subtraction and noise type for unvoiced speech segregation. In *Proceedings of ICASSP*, pages 4425–4428, 2009. 21

[43] Y. Hu and P. C. Loizou. A Comparative Intelligibility Study of Single-Microphone Noise Reduction Algorithms. *Journal of the Acoustical Society of America*, 122:1777–1786, 2007. 20, 118

[44] Y. Hu and P. C. Loizou. Techniques for estimating the ideal binary mask. In *Proceedings of 11th International Workshop on Acoustic Echo and Noise Control*, 2008. 2, 20

[45] X. Huang, A. Acero, and H.-W. Hon. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall, Inc., 2001. 6, 7, 15

[46] C. Hummersone, R. Mason, and T. Brookes. Ideal binary mask ratio: a novel metric for assessing binary-mask-based sound source separation algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2039–2045, 2011. 118

[47] P. I. M. Johannesma. The pre-response stimulus ensemble of neurons in the cochlear nucleus. In *Proceedings of the Symposium on Hearing Theory*, pages 58–69, Eindhoven, Netherlands, June 1972. 23

[48] K. Johnson. *Acoustic and Auditory Phonetics*. Blackwell Publishing, Malden, MA, 2nd edition, 2003. 7

[49] J.D. Johnston. Estimation of Perceptual Energy Using Noise Masking. In *Proceedings of ICASSP*, pages 2524–2427, 1988. 18

[50] C. Kim and R. M. Stern. Power-normalized cepstral coefficients for robust speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 2013. 1, 17

[51] W. Kim and J. H. L. Hansen. Feature compensation in the cepstral domain employing model combination. *Speech Communication*, 51:83–96, 2009. 90

[52] W. Kim and J. H. L. Hansen. Mask estimation employing posterior-based representative mean for missing-feature speech recognition with time-varying background noise. In *Proceedings of ASRU*, pages 194–198, Meraono, Italy, 2009. 89

[53] W. Kim and J. H. L. Hansen. Missing-feature reconstruction by leveraging temporal spectral correlation for robust speech recognition in basckground noise conditions. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8):2111–2120, November 2010. 32, 45

[54] W. Kim and J. H. L. Hansen. A novel mask estimation method employing posterior-based representative mean estimate for missing-feature speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(5):1434–1443, July 2011. 56, 87, 90, 92

[55] V. Kolmogorov and R. Zahib. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:147–159, 2004. 98

[56] J. D. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabalistic models for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning*, pages 282–289, San Francisco, 2001. 98

[57] T.-W. Lee and K. Yao. Speech Enhancement by Perceptual Filter with Sequential Noise Parameter Estimation. In *Proceedings of IEEE ICASSP*, volume 11, pages 466–469, 2004. 18

[58] C. J. Leggetter and P. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech and Language*, 9(2):171–185, 1995. 17

[59] R. G. Leonard. A database for speaker-independent digit recognition. In *Proceedings IEEE International Conference on Speech Acoustics and Signal Processing*, pages 111–114, 1984. 30, 34, 37, 58

[60] J. Li, L. Deng, D. Yu, Y. Gong, and A. Acero. High-performance hmm adaptation with joint compensation of additive and convolutive distortions via vector taylor series. In *Proceedings of ASRU*, pages 65–70, 2007. 17

[61] N. Li and P. C. Loizou. Factors influencing intelligibility of binary-masked speech: Implications for noise reduction. *Journal of the Acoustical Society of America*, 123:1673–1682, 2008. 64

[62] Y. Li and D. L. Wang. On the optimality of binary time-frequency masks. *Speech Communication*, 51:230–239, 2009. 18

[63] P.C. Loizou. *Speech Enhancement: Theory and Practice*. CRC Press, 2007. 28, 35

[64] R. F. Lyon. A computational model of filtering, detection, and compression in the cochlea. In *Proceedings of ICASSP*, pages 1282–1285, Paris, May 1982. 23

[65] J.-B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, The Google Books Team, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. A. Nowak, and E. L. Aiden. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, 2011. 13

[66] G. Miller. Decision units in the perception of speech. *Transactions on Information Theory*, 8(2):81–83, 1962. 12

[67] S. Molau, F. Hilger, and H. Ney. Feature space normalization in adverse acoustic conditions. In *Proceedings of ICASSP*, volume 1, pages 656–659, 2003. 8

[68] B. C. J. Moore. *An Introduction to the Psychology of Hearing*. Academic Press, Inc., San Diego, CA, 5th edition, 2003. 19

[69] P. J. Moreno, B. Raj, and R. M. Stern. A vector taylor series approach for environment-independent speech recognition. In *Proceedings of ICASSP*, pages 733–736, 1996. 17

[70] J. Morris and E. Fosler-Lussier. Conditional random fields for integrating local discriminative classifiers. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(3):617–628, March 2008. 99

[71] A. Narayanan and D. L. Wang. Robust speech recognition from binary masks. *Journal of Acoustical Society of America Express Letters*, 128:EL217–222, 2010. 20

[72] A. Narayanan, X. Zhao, and D. L. Wang. Uncertainty decoding for robust speech recognition using multiple speech priors. In *Proceedings of Interspeech*, 2010. 118

[73] A. M . Noll. Cepstrum pitch determination. *Journal of Acoustical Society of America*, 47:293–309, 1967. 7

[74] A. Papoulis and S. U. Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 4th edition edition, 2002. 70

[75] N. Parihar and J. Picone. Analysis of the aurora large vocabulary extensions. In *Proceedings of Eurospeech*, volume 4, pages 337–340, Geneva, Switzerland, September 2003. 13, 34, 37, 44, 65

[76] R. D. Patterson, M. H. Allerhand, and C. Giguere. Time-domain modeling of peripheral auditory processing: A modular architecture and a software platform. *Journal of Acoustical Society of America*, 98(4):1890–1894, 1995. 23

[77] R. D. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice. An efficient auditory filterbank based on the gammatone function. Technical report, MRC Applied Psychology Unit, Cambridge, 1987. 6, 23

[78] D. Paul and J. Baker. The design of wall street journal-based CSR corpus. In *Proceedings of International Conference on Spoken Language*, pages 899–902, Banff, Alberta, Canada, October 1992. 37, 44

[79] D. Povey, M. Hannemann, G. Boulianne, L. Burget, A. Ghoshal, M. Janda, M. Karafiat, S. Kombrink, P. Motlicek, Y. Qian, K. Riedhammer, K. Vesel, and N. T. Vu. Generating exact lattices in the wfst framework. In *Proceedings of ICASSP*, 2012. 15

[80] P. Price, W. M. Fisher, J. Bernstein, and D. S. Pallett. The darpa 1000-word resource management database for continuous speech recognition. In *Proceedings of IEEE ICASSP*, 1988. 33

[81] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77:257–286, 1989. 4, 10, 14

[82] A. Ragni and M. J. F. Gales. Structured discriminative models for noise robust continuous speech recognition. In *Proceedings of IEEE International Conference on Speech Acoustics and Signal Processing*, pages 4788–4791, 2011. 103

[83] B. Raj, M. L. Seltzer, and R. M. Stern. Reconstruction of missing features for robust speech recognition. *Speech Communication*, 43:275–296, 2004. 1, 2, 30, 31, 37, 57, 65, 116

[84] S. J. Rennie and P. L. Dognin. Beyond linear transforms: Efficient non-linear dynamic adaptation for noise robust speech recognition. In *Proceedings of Interspeech*, 2008. 30

[85] M. Van Segbroeck and H. Van Hamme. Advances in missing feature techniques for robut large-vocabulary continuous speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 19(1):123–137, January 2011. 31, 103

[86] S. Seneff. A joint synchrony/mean-rate model of auditory speech processing. *Journal of Phonetics*, 16:55–76, 1988. 23

[87] Y. Shao, Z. Jin, and D. L. Wang. An auditory-based feature for robust speech recognition. In *Proceedings of ICASSP*, 2009. 6

[88] S. Srinivasan. *Integrating computational auditory scene analysis and automatic speech recognition*. PhD thesis, The Ohio State University, 2006. 38

[89] S. Srinivasan, N. Roman, and D. L. Wang. Binary and ratio time-frequency masks for robust speech recognition. *Speech Communication*, 48:1486–1501, 2006. 30, 38, 57

[90] S. Srinivasan and D. L. Wang. A schema-based model for phonemic restoration. *Speech Communication*, 45:63–87, 2005. 110

[91] S. Srinivasan and D. L. Wang. Transforming binary uncertainties for robust speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2130–2140, 2007. 31, 32, 34, 37, 45, 46

[92] S. Srinivasan and D. L. Wang. Robust speech recognition by integrating speech separation and hypothesis testing. *Speech Communication*, 52:72–81, 2010. 38, 57, 58, 103

[93] V. Stouten, H. Van Hamme, and P. Wambacq. Joint removal of additive and convolutional noise with model-based feature enhancement. In *Proceedings IEEE International Conference on Speech Acoustics and Signal Processing*, volume 1, pages 949–952, 2004. 103

[94] M. K. Tanenhaus, M. J. Spivey-Knowlton, M. K. Eberhard, and J. C. Sedivy. Integration of visual and linguistic information in spoken language comprehension. *Science*, 268(5217):1632–1634, 1995. 12

[95] A. P. Varga, H. J. M. Steeneken, M. Tomlinson, and D. Jones. The NOISEX-92 Study on the Effect of Additive Noise on Automatic Speech Recognition. Technical report, Speech Research Unit, Defense Research Agency, Malvern, UK, 1992. 39, 78, 94

[96] D. L. Wang. On ideal binary mask as the computational goal of auditory scene analysis. In P. Divenyi, editor, *Speech separation by humans and machines*, pages 181–197. Kluwer Academic, Norwell MA, 2005. 1, 19

[97] D. L. Wang and G. Brown, editors. *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Wiley-IEEE Press, 2006. 1, 6, 18, 20, 23, 28, 35, 57

[98] D. L. Wang and G. J. Brown. Separation of speech from interfering sounds based on oscillatory correlation. *IEEE Transactions on Neural Networks*, 10:684–697, 1999. 19

[99] D. L. Wang, U. Kjems, M. S. Pedersen, J. B. Boldt, and T. Lunner. Speech perception of noise with binary gains. *The Journal of the Acoustical Society of America*, 124:2303–2307, 2008. 20, 54, 59

[100] D. L. Wang, U. Kjems, M. S. Pedersen, J. B. Boldt, and T. Lunner. Speech intelligibility in background noise with ideal binary time-frequency masking. *The Journal of the Acoustical Society of America*, 125:2336–2347, 2009. 1, 20, 28, 54

[101] M. Weintraub. The GRASP sound separation system. In *Proceedings of IEEE ICASSP*, pages 18A.6.1–4, 1984. 19

[102] M. Westphal. The use of cepstral means in conversational speech recognition. In *In Proceedings of the European Conference on Speech Communication and Technology*, pages 1143–1146, 1997. 1, 8

[103] J. Woodruff, R. Prabhavalkar, E. Fosler-Lussier, and D. L. Wang. Combining monaural and binaural evidence for reverberant speech segregation. In *Proceedings of Interspeech*, Makuhari, Japan, 2010. 98

[104] S. Young. A Review of Large-Vobaculary Continuous-Speech Recognition. *IEEE Singal Processing*, 96:45–57, 1996. 4

[105] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book*. Cambridge University Publishing Department, 2002. 15, 35, 39, 44, 64

[106] G. Zweig and P. Nguyen. A segmental crf approach to large vocabulary continuous speech recognition. In *Proceedings of ASRU*, pages 152–157, 2009. 99