JAMES COOK UNIVERSITY

SCHOOL OF ENGINEERING

EG4011/2

Practical Speech Enhancement in Babble using Non-negative matrix Factorisation

Ashley Gillman
Bachelor of Engineering (Electrical)

Thesis Proposal

May, 9th 2014

Table of Contents

Chapter 1: Introduction 1
1.1 Research Questions 1
1.2 Scope 2

Chapter 2: Literature Review 3
2.1 Babble and other Speech-Related Noise 3
2.2 Speech Enhancement 4
2.3 Modelling Babble 6
2.3.1. For Generation 6
2.3.2. For Analysis and Enhancement 6
2.3.3. Separating Speech from Babble 6
2.4 Methods for Evaluation of Speech Enhancement 7
2.4.1. Human Recognition 7
2.4.2. Machine Recognitions 8
2.5 Subspace Methods by Decomposition 9
2.5.1. Non-negative Matrix Factorisation and Speech 10
2.6 Assessment of Methods of Factorisation 11
2.6.1. Objective Functions 11
2.6.2. Multiplicative Update Algorithms 12
2.6.3. Alternating Least Squares Algorithms 13
2.6.4. Exemplar-Based NMF 14
2.6.5. Constrained NMF 14
2.6.6. General NMF approaches to Source Separation 15
2.7 Summary 16

Chapter 3: Methodology 17
3.1 Implementation Infrastructure 17
3.2 Algorithms 17
3.2.1. Algorithm I – Non-negative Matrix Factorisation with Priors 17
3.2.2. Algorithm II – Non-negative Matrix Factorisation of Phonemes with Priors 18
3.2.3. Reducing the Training Requirements 19
3.3 Test Data 20
3.4 Evaluation Measures 21
3.5 Project Management 22
3.6 Resource Management and Cost Analysis 22

References 23

List of Figures

Figure 2-1 – Some classifications of Speech Enhancement Methods 5
Figure 2-3 – Evaluation methods used in literature 9
Figure 2-4 – Mathematical decomposition techniques 9
Figure 2-5 – NMF factorising speech into components [17]. 11
Figure 2-6 – Common NMF speech implementation 15

List of Tables

Table 2-1 – Speech-related noise types.....3
Table 2-2 – Subjective classification of babble noise vs. SSN [6] 4
Table 3-1 – Difficulty Classification 21

List of Symbols

In this thesis, matrices are denoted by capitalised letters. Subscript notation on matrices using commas denotes indexation (i.e. $A_{i,j}$ represents the ith row and jth column) whereas using the multiplication symbol, "×", denotes a size definition (i.e. $A_{i \times j}$ is an i-by-j matrix).

$\otimes$	Elementwise multiplication
$\oslash$	Elementwise division
$\hat{.}$	A hat is used to indicate an estimated value
/./	Forward slashes are used to represent phonemes
c	Component base index
D_{KL}	Kullback-Leibler divergence
f	Frequency bin index
Λ	Covariance
n	The length of the given vector
n_c	The total number of component bases
n_f	The total number of frequency bins
n_s	The total number of time frames
p	Fourier transform vector of a phoneme slice
φ	Objective function or cost function
s	Time frame index
V	Spectral component matrix, one of the decompositions formed through the non-negative matrix factorisation procedure. Rows represent frequency bins and columns represent the component bases
W	Activation matrix, one of the decompositions formed through the non-negative matrix factorisation procedure. Rows represent the component bases and columns represent time frames
x	A signal or recording, either with or without noise present
X	Short-time Fourier transform of a signal or recording, either with or without noise present. Rows represent frequency bins and columns represent time frames
y	An observed signal or recording
Y	Short-time Fourier transform of an observed signal or recording. Rows represent frequency bins and columns represent time frames

List of Abbreviations

ALS	Alternating Least Squares
ASR	Automated Speech Recognition
DSP	Digital Signal Processing
HI	Hearing Impaired
HMM	Hidden Markov Model
HR	Human Recognition
ITU	International Telecommunication Union
NMF	Non-negative Matrix Factorisation
PESQ	Perceptual Evaluation of Speech Quality
SNR	Signal-to-Noise Ratio
SoI	Speaker of Interest
SSN	Speech-Shaped Noise
WSJ	Wall Street Journal

Chapter 1: Introduction

The "cocktail party problem" was first posed in 1953 by Cherry [1], where the human ability, or often difficulty, to hear speech in the presence of multiple speakers was noted. After analysis into the complexity of the problem, it is amazing that humans have the ability to hear over one-another at all!

The cocktail party problem refers to the problem of recognising speech in the presence of babble. A number of speakers are present, and each can be distinguished individually. This has been noted as an extremely difficult task in speech analysis and enhancement. With the rise of modern technology and the desire to incorporate alternative human-machine interfaces, the motivation to improve Automated Speech Recognition Systems (ASR) has increased. Additionally, the problem still exists of aiding human understanding in such situations, e.g. hearing aid systems or telecommunications systems.

One method of subspace analysis that has shown promising results is that of Non-negative Matrix Factorisation (NMF). This is a relatively new method of decomposition proposed by Lee and Seung [2], with applications to spectral analysis due to the non-negativity constraints of NMF and the non-negative nature of spectral magnitude data. The components of a desired signal and be learned and extracted from a signal. Babble filtering systems are required to be trained to recognise the individual speaker, which is often a difficult process and a practical limitation in these systems.

A number of different challenges have been held with the motivation of improving the performance of ASR systems under difficult noise conditions [3-5]. Entries into such competitions can be broadly categorised into two categories, those that perform recognition themselves, and those that clean the signal and supply a cleaned signal to a standardised recogniser. Algorithms that fall into the latter category have a possible additional application: improving intelligibility for human listeners. It is these algorithms that are of interest in this thesis.

1.1 Research Questions

The aim of this thesis is to address the following research questions:

1. Are good enhancement algorithms effective for both human listeners and machine listeners? Can a generic and practical speech enhancement algorithm find application in signal enhancement and ASR?

2. Can the results be improved by modifying algorithms to concentrate the focus on recognition of the desired speakers voice?
3. Can the practicality of existing algorithms be improved to allow applications in end-user hardware?

1.2 Scope

The scope of this thesis is to develop a general and practical speech enhancement algorithm with applications in a number of areas, including enhancement for human listeners and for ASR systems. Such an algorithm should be efficient, have low requirements, and operate on monaural recordings.

The scope does not encompass specific applications, and thus does not consider special requirements beyond normal hearing for hearing aid and cochlear implant listeners. It is assumed that the performance increases for normal hearing listeners will be applicable to impaired hearing listeners. Further studies would be required to ensure this is the case.

Chapter 2: Literature Review

This literature review aims to outline some of the key concepts involved in this thesis, and to summarise progress on similar endeavours made thus far.

2.1 Babble and other Speech-Related Noise

In signal analysis, noise is defined as an unwanted signal. In acoustics, a wide range of types of noise may interfere with a signal. This noise originates from somewhere within the environment: from a stationary source, or a non-stationary source. Generally this noise is additive, meaning the mixed signal can be considered as the desired signal plus noise.

In this thesis the desired signal, or Speaker of Interest (SoI) signal, is considered to be a speech signal and the additive noise signal (or masker) is considered to be a number of competing speech signals. Noise originating from speakers can be classified by the number of speakers as Speech-Shaped Noise (SSN), babble noise or competing speaker noise, as defined in Table 2-1.

Table 2-1 – Speech-related noise types

Type of Speech-Related Noise	Definition [6]	Steady-state or time-varying
SSN	A diffused background rumble, where individual conversations or speakers are not distinguishable.	Steady-state
Babble Noise	Individual speakers can be heard and at times, individual words can also be heard.	Time-varying
Competing Speaker	There are two speakers present.	Time-varying

Babble noise is often considered the most difficult noise condition for speech processing due to the fact it is very similar in structure to the desired clean signal, and due to the time-variant structure (unlike SSN) [6].

The exact distinction between babble and SSN is under-defined and subjective. Table 2-2 below shows test results by Krishnamurthy and Hansen [6] where subjects were asked to listen to a recording of a number of speakers and classify whether they could identify individual speakers. It was found that for four to six speakers the classification of whether noise was babble or speech-shaped varied from person-to-person, even for the same recording.

Table 2-2 – Subjective classification of babble noise vs. SSN [6]

Number of Speaker in Babble	Percentage of Speakers Identifying Recording as Babble	Percentage of Speakers Identifying Recording as Speech-Shaped
≤3	100%	0%
4	66%	34%
5	18%	82%
6	27%	73%
≥7	0%	100%

Human listeners compensate for babble and competing speaker noise by exploiting the time-modulated property of the noise. Listeners focus on the target speech during moments of low noise levels to piece together the target's message. Therefore, human recognition in modulated noise is better than in steady-state noise. However, the opposite is true for a machine: steady state noise conditions are easier to handle than modulated noise with Digital Signal Processing (DSP) techniques [7]. Of course, the level of improvement directly depends on the speech enhancement technique used.

2.2 Speech Enhancement

Speech enhancement has various practical applications, including:

- improving hearing with hearing aids and cochlear implants [8, 9],
- increasing the accuracy of ASR systems,
- denoising of telecommunications systems
- and refining recorded audio [10].

Due to the many application areas, various algorithms may be used under different contexts in order to balance the condition-specific performance, calculation efficiency, flexibility and ease of implementation.

Figure 2-1 illustrates some of the many variations in methods for speech enhancement, with data extracted from [9, 11-13]. Some methods provide better performance than others, but at the expense of practicality. For example, recognition can be improved under non-stationary noise by using a microphone array, however such a set up is more expensive.

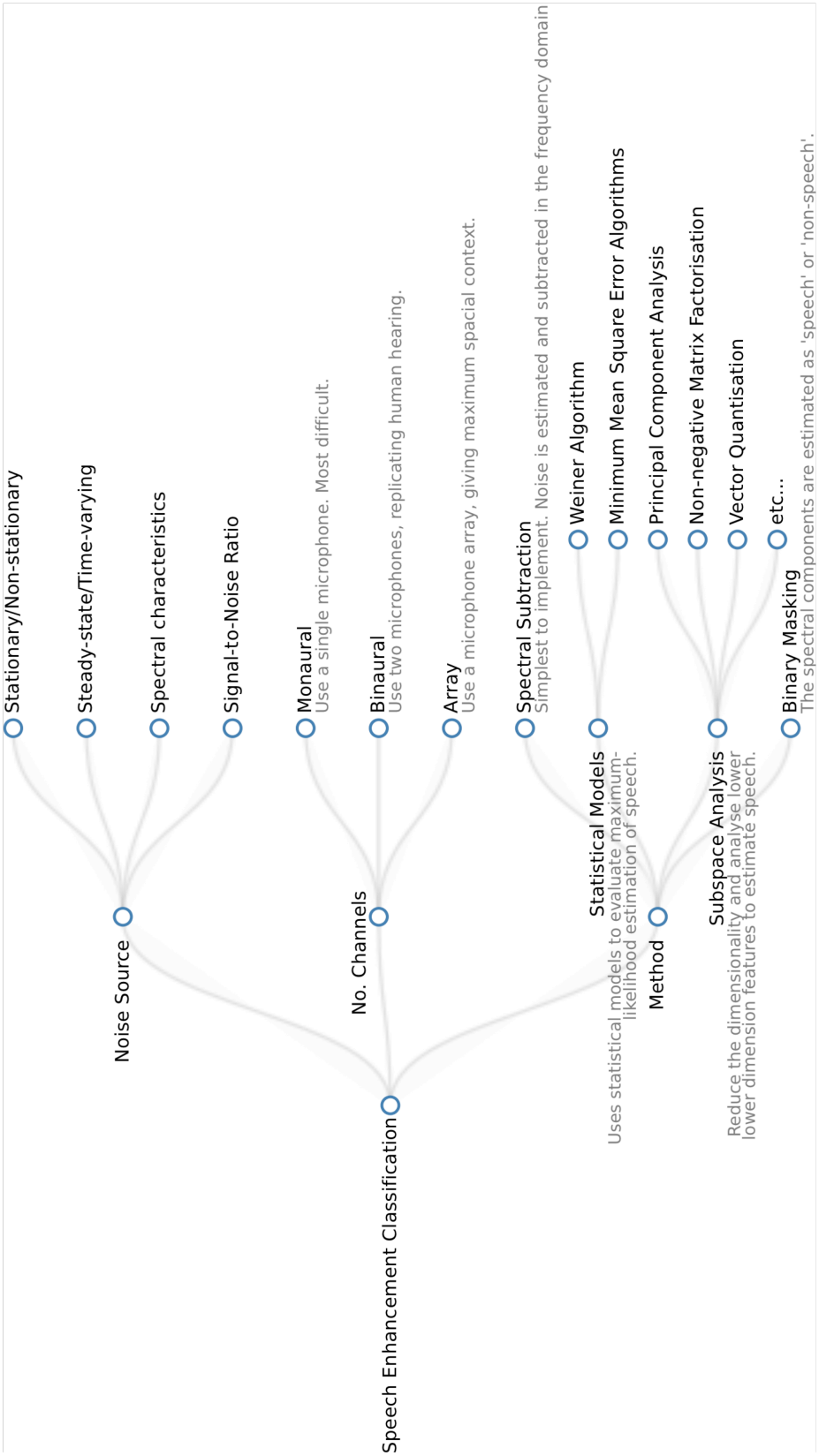

Figure 2-1 – Some classifications of Speech Enhancement Methods

2.3 Modelling Babble

In a multi-speaker environment, and ignoring other forms of noise, a signal can be considered a sum of multiple speakers:

$$X_{Babble} = \sum X_{Speakers} \quad \textbf{(2-1)}$$

(2-1) follows directly from the definition of babble. The number of speaker signals used may vary, but to align with with the definition for babble should be between three and six.

2.3.1. For Generation

It is generally desirable in testing to be able to present noise at different Signal-to-Noise Ratios (SNRs). Rather than recording noise in a range of conditions directly, it is more convenient to have separate speech and noise and to model the speech with noise signal. Thus, the effects of babble are often modelled rather than directly recorded. This type of babble can be classed as synthetic babble, and may or may not accurately represent real babble [6].

The most common method of creating synthetic babble is to use a number of speech recordings, adding them together to form babble. This presents a number of issues. Firstly, real conversations are not simply a number of voices speaking simultaneously. Rather, conversations are dynamic, where there is generally one speaker but occasionally more or none [6]. Secondly, synthetic babble generally does not take into effect environmental effects, such as reverberation, generally present alongside babble [6].

2.3.2. For Analysis and Enhancement

For analysis, it is generally preferable to model a signal as a sum consisting of a desired component and a noise component. The babble model may exploit (2-1), such that a single voice is modelled and extended to multiple speakers [14], whereas other methods leave the model for babble as a generic noise model [15]. The advantage of the latter is that the same model may be applicable to other forms of noise.

2.3.3. Separating Speech from Babble

When the cocktail party problem was originally proposed, five factors were identified as to how the human brain may differentiate between the sources [1], shown in Figure 2-2.

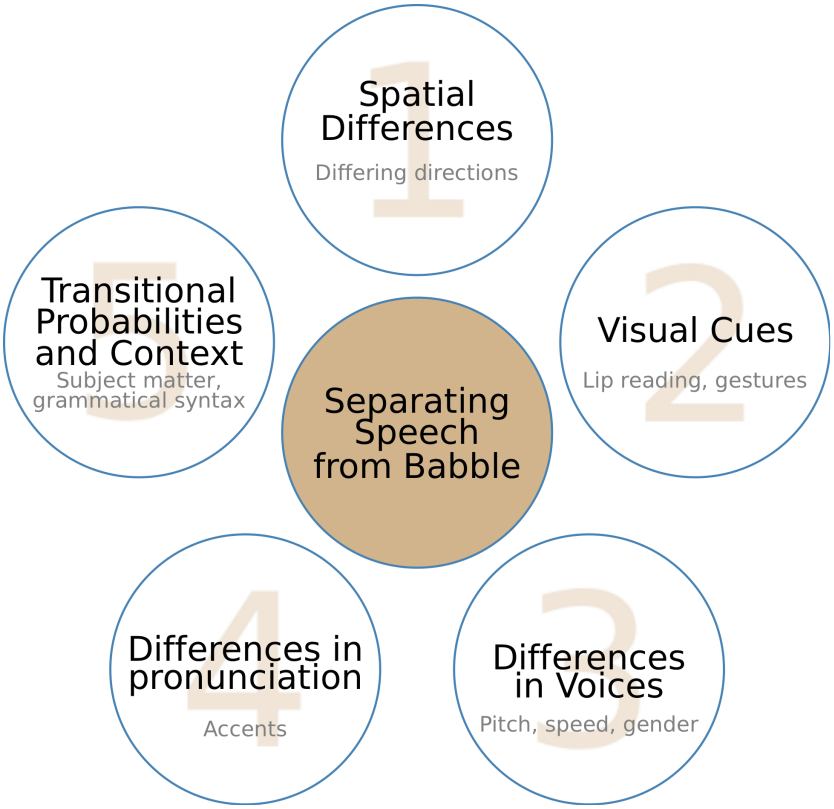

Figure 2-2 – Differentiating Speech from Babble

Algorithms taking advantage of spatial differences and visual cues, although effective [16], require multiple channels and visual data. Subspace methods take advantage of differences in voices and differences in pronunciation. HMM models for speech recognition also take advantage of transitional probabilities [17].

2.4 Methods for Evaluation of Speech Enhancement

Methods for evaluating the intelligibility of speech, and by extension speech enhancement, can be classified by the type of recognition they measure: ASR vs. HR. This is necessary since ASR systems often recognise speech through a relatively low dimension feature vector, whereas the human brain is far more sensitive to the signal. An algorithm for enhancement may indeed improve the intelligibility for an ASR system, but side effects may be distracting or distortive to a human listener.

2.4.1. Human Recognition

Intelligibility of a speech signal by humans is difficult to accurately and quantitatively measure. The most obvious measure is to have human test subjects listen to signals and judge which are more intelligible. In order to achieve reputable, repeatable and accurate results a large test is required with a large number of test subjects and recordings.

The International Telecommunication Union's (ITU's) standard for such tests is the Mean Opinion Score (MOS) [18]. The test is designed to give an indication of the quality of telecommunication transmission, but is applicable as a measure of speech intelligibility. The MOS results are a score calculated by the mean of the listeners' scores in the range one to five, with one representing the worst quality and five representing the highest quality of intelligibility.

Perceptual Evaluation of Speech Quality (PESQ) is the ITU's standard for evaluation of objective speech quality [19]. This method provides an algorithm which estimates the improvement quality by comparing a system's input and output [20]. Results show PESQ scores give consistent and reliable estimates on human perception, although are not directly comparable with MOS scores [21].

Other methods for measuring perceptual quality include long-term SNR, segmental SNR, weighted spectral slope, log-likelihood ratio, Itakura-Saito distance, cepstrum distance, Spectral Distortion (SD), Source-to-Distortion Ration (SDR) and variations [22].

2.4.2. Machine Recognitions

The most common technique to evaluate machine recognition quality of speech is to run an ASR algorithm on the signal and perform a comparison of the Word Recognition Rate (WRR). It should be noted that this method is somewhat limited, in that different ASR algorithms, which may implement different feature vectors, may perform differently. As such, results from different ASR algorithms are not directly comparable.

Algorithms in literature tend to limit evaluation methods to either human recognition or machine recognition methods, as seen in Figure 2-3, showing the evaluation measures used in various papers. This data was gathered from the following papers [14, 15, 23-27].

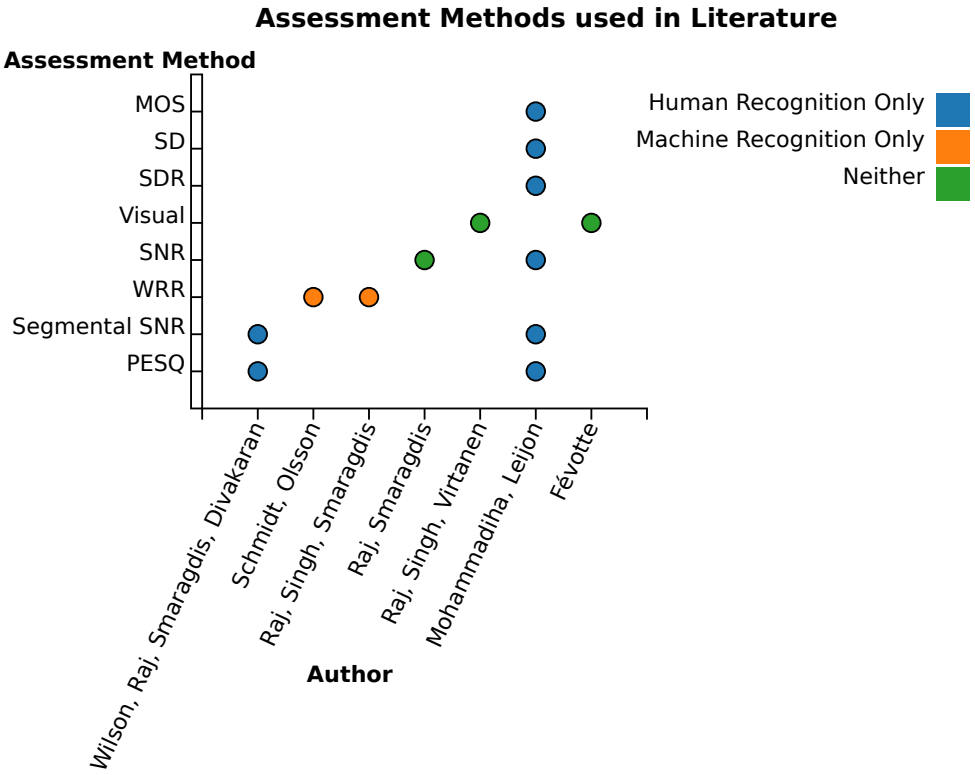

Figure 2-3 – Evaluation methods used in literature

2.5 Subspace Methods by Decomposition

Subspace methods focus on reducing the dimensionality of a system. This is done by decomposition or factorisation of a signal into lower dimensional systems. Many techniques exist for decomposing a signal, some of which are outlined in Figure 2-4.

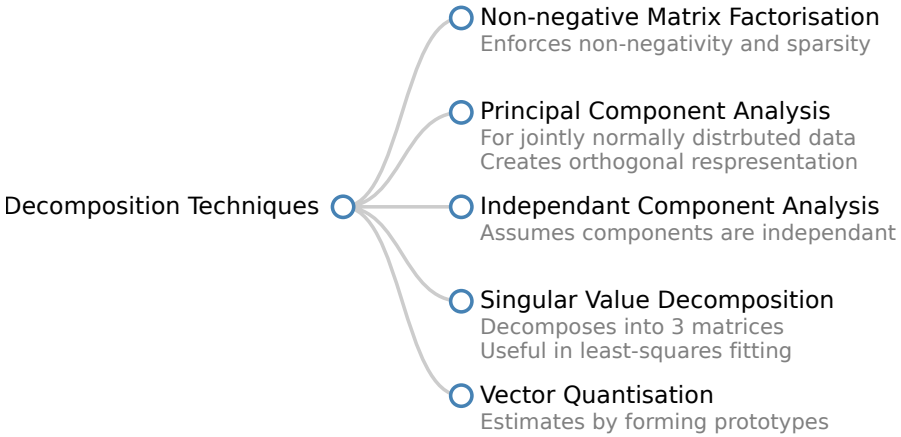

Figure 2-4 – Mathematical decomposition techniques

These techniques factorise a matrix into two matrix factors of the form:

$$C_{i\times j} \approx B_{i\times k}A_{k\times j} \tag{2-2}$$

Often $k \ll i$ and $k \ll j$, meaning the factorised form may be considered a more compact form.

Non-negative Matrix Factorisation (NMF), previously known as positive matrix factorisation, was popularised by Lee and Seung [2] as a method lending itself to breaking objects into their composing parts The distinguishing characteristic of NMF is that the product matrix, along with both its factors, are non-negative, i.e. $C \geq 0$, $B \geq 0$ and $A \geq 0$. NMF has been demonstrated to be an effective method in dividing images into parts [2, 28]. Furthermore, NMF also has been shown to be useful on signals, specifically as a source-separation algorithm [14, 15, 17].

2.5.1. Non-negative Matrix Factorisation and Speech

NMF can be used in signal analysis, such that [2, 29]:

$$X_{n_f \times n_s} \approx V_{n_f \times n_c} W_{n_c \times n_s} \quad \textbf{(2-3)}$$

where:

- X represents the signal with rows of frequency bins results from a Fourier transform and columns of time segments;
- V is the spectral components with columns representing typical spectral vectors;
- W is the activation matrix, where each row represents the activation levels of components at given times;
- n_f is the number of Frequency bins;
- n_s is the number of samples; and
- n_c is the number of components.

Figure 2-5 shows a visual representation of this factorisation. It can be seen that the columns of the spectral component matrix, V, represent approximations of the spectral components, and that the activation matrix, W, is sparse.

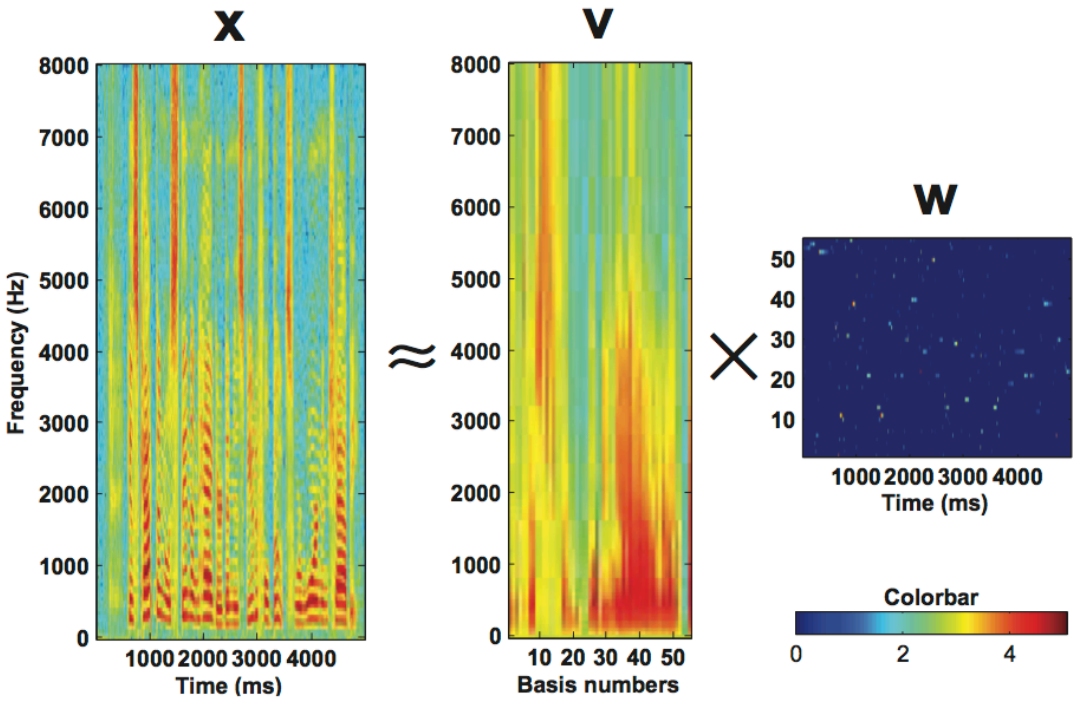

Figure 2-5 – NMF factorising speech into components [17].

2.6 Assessment of Methods of Factorisation

2.6.1. Objective Functions

NMF algorithms aim to minimise or maximise an objective function. The objective function is designed with an intent that optimisation will ensure the approximation in (2-3).

The common choice is to minimise the square error, or Euclidean distance between X and VW seen in (2-4) [30].

$$F = \|X - VW\|^2 = \sum_{f=1}^{n_f} \sum_{s=1}^{n_s} \left[X_{f,s} - (VW)_{f,s}\right]^2 \tag{2-4}$$

Another standard objective function based on the Kullback-Leibler divergence of VW from X

$$D_{KL}(X \parallel V_{all} W_{all}) \tag{2-5}$$

was proposed by Lee and Seung [2] and is given as (2-6). Here, the objective is maximisation of the divergence objective function.

$$F = \sum_{f=1}^{n_f} \sum_{s=1}^{n_s} \left[X_{f,s} \log(VW)_{f,s} - (VW)_{f,s}\right] \tag{2-6}$$

Often a modified objective function is used. This often adds a cost function in order to achieve auxiliary constraints [31]. Some examples of this are further discussed below. The objective function is implemented using a series of update functions. The update functions are iteratively processed until the approximation of VW to X is sufficient.

2.6.2. Multiplicative Update Algorithms

Updates are often implemented multiplicatively, since if all initial matrices are non-negative the final matrices will also implicitly be non-negative. This prevents the need to explicitly enforce non-negativity. For the Euclidean distance objective function, the multiplicative update rules are [30]

$$V_{f,c} \leftarrow V_{f,c} \otimes (XW^T) \oslash (VWW^T) \tag{2-7}$$

and

$$W_{c,s} \leftarrow W_{c,s} \otimes (V^T X) \oslash (V^T VW) \tag{2-8}$$

where $\otimes$ and $\oslash$ are the elementwise multiplication and division operators. Alternatively, if the divergence objective function is to be used,

$$V_{f,c} \leftarrow V_{f,c} \sum_s \frac{X_{f,s}}{(VW)_{f,s}} W_{c,s} \tag{2-9}$$

and

$$W_{c,s} \leftarrow W_{c,s} \sum_s V_{f,c} \frac{X_{f,s}}{(VW)_{f,s}} \tag{2-10}$$

provide the appropriate update rules [2]. Additionally, a third update rule [2],

$$V_{f,c} \leftarrow \frac{V_{f,c}}{\sum_f V_{f,c}} \tag{2-11}$$

was included to normalise columns in the component matrix. This is included to prevent continuously updating V by a constant and W by its inverse, which would otherwise be a valid update since VW would be identical.

In addition, equivalent additive expressions can be formulated, known as the gradient descent algorithms [30, 31]. In this thesis, these forms are considered as equivalent [31] and thus are not considered separately.

The multiplicative update method of NMF has been used extensively in literature with good results [15, 24]. However, it has been shown that the proof given by Lee and Seung [30], attempting to prove the convergence of the multiplicative update rules (2-7) to (2-11), was incomplete [31, 32]. The above equations may become stuck at stationary points and do not always converge to the desired objective function [31-33]. When an element in V or W becomes zero it becomes "locked" and must remain as zero, and inherent flaw of the multiplicative update functions [34]. The following class of algorithms allow flexibility in this respect.

2.6.3. Alternating Least Squares Algorithms

Alternating Least Squares (ALS) algorithms are based on the initial proposed algorithms by Paatero and Tapper [29]. These algorithms are based on the alternating variables method (or coordinate descent method) [35]. In these algorithms, first V is fixed and W is calculated using a least squares method, then W is fixed and V is calculated in a similar manner. These two steps are repeated until an adequate approximation is achieved [34].

The least squares method used varies between ALS algorithms. ALS can be generally classified by whether a non-negative least squares method is used; or a general least squares method where any negative values are subsequently set to zero. Non-negative least squares methods converge to a local minimum successfully, however the calculation cost per iteration is far higher despite attempts to improve performance [31, 34, 36-38]. General least squares methods may converge to a saddle point as opposed to a minima. This is generally still preferred under practical conditions due to the speed. Performance can also be improved by using a good initialisation.

Typical ALS algorithms [31] using general least squares method resemble the following:

1. Initialise V
2. Repeat the following for a predefined number of iterations:
    a. Solve for W using (2-13).
    b. Set any negative elements in W to zero.
    c. Solve for V using (2-13).
    d. Set any negative elements in V to zero.

$$V^T V W = V^T X \quad \textbf{(2-12)}$$

$$W W^T V^T = W X^T \quad \textbf{(2-13)}$$

The algorithms presented thus far present a common weakness in that the update rules for the component matrix and the activation matrix are the same (except that in some cases the component matrix also has a normalisation update). This means there is little control over which information is captured in the component matrix and which is captured in the activation matrix.

The following algorithms are variations that attempt to provide further control over ensuring that the components are completely captured in the component matrix.

2.6.4. Exemplar-Based NMF

This variation on the multiplicative update algorithm can be made where it is expected that the components of the signal present themselves individually, i.e. the signal is assumed to be a series of components as opposed to combinations. Here, the component matrix components are drawn from the desired signal. This leaves only the activation matrix to be calculated, so only one update rule is required, (2-10) [26].

This is the case for clean speech signals, where bases are taken to be phonemes [26]. Taking advantage of this, phonemes may be drawn randomly from speech and used to form the spectral component matrix. The advantage of this method is that meaning is brought to the components. This allows activations of a given component matrix to be compared with those of another.

2.6.5. Constrained NMF

Another method to ensure component information is captured within the component matrix is to introduce sparseness constraints on the activation matrix. By it's nature, the activation matrix should be sparser than the component matrix. Through the non-negative constraint of NMF, sparseness of the activation matrix (and indeed, also the component matrix) naturally occurs [34, 39]. However, it can be desirable to allow explicit control over the sparseness, especially in speech applications [23].

Hoyer [40] proposed an algorithm based on the multiplicative update algorithm in [2] with an introduced cost function to aid in control of the sparseness. This algorithm was further improved in [39] where the explicit control of sparseness was introduced. This was achieved by defining the sparseness measure [39]:

$$sparseness(x) = \frac{\sqrt{n} - (\sum |x_i|)/\sqrt{\sum x_i^2}}{\sqrt{n} - 1} \quad \textbf{(2-14)}$$

Where n is the length of vector x. The developed algorithm is then based on the Euclidean distance objective function, employing a combination of multiplicative update functions (§2.6.2) and gradient descent functions. However, the theory developed is applicable to many types of objective functions and implementation algorithms [41]. Similarly, the cost functions introduced need not be specific to sparsity, but can be applied to many auxiliary constraints, such as enforcing smoothness [27].

2.6.6. General NMF approaches to Source Separation

There are a number of methods to construct a clean-speech estimation using NMF. The particular methods used largely depend on the specifics of the model used. However, the general concept is the same. It is known that X is approximated as $V \times W$ (2-3) and therefore we can we can reconstruct a desired signal as:

$$\hat{X}_{Desired} = V_{Desired} \times \hat{W}_{Desired} \quad \textbf{(2-15)}$$

Common algorithms form a component matrix for the desired speaker, and another for the anticipated noise [15, 23, 24]. These are concatenated, and the appropriate NMF algorithms performed. Afterwards, the speech-specific components can be separated again and used to estimate the clean speech.

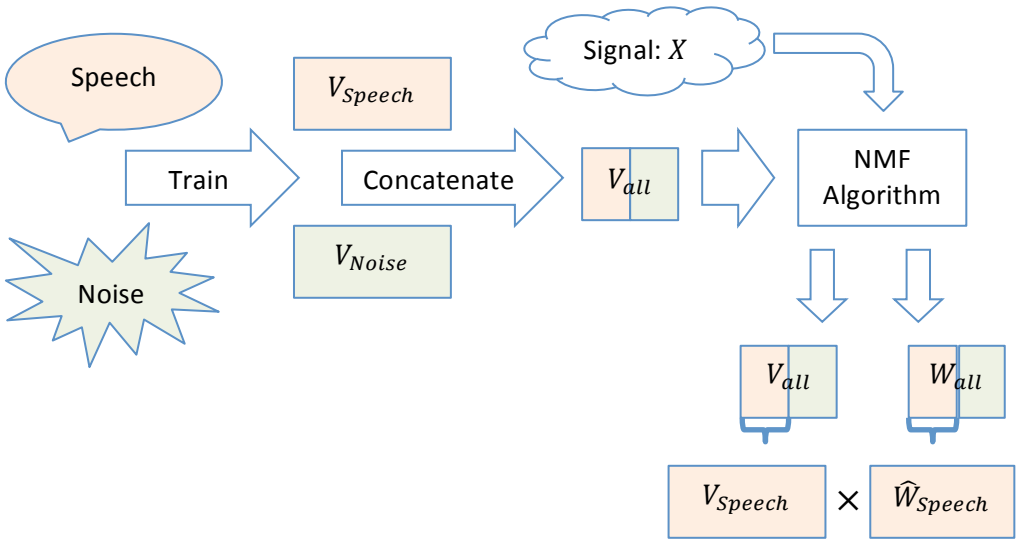

Figure 2-6 – Common NMF speech implementation

Limitations of these models mostly belong to the training, meaning required *a priori* knowledge. One issue is the required resources for training, which can be upwards of an hour worth of recordings [14, 26]. Another issue is that these algorithms are only effective for noise types that V_{noise} has been trained on, and thus have reduced flexibility.

Raj, et al. [26] proposed a different solution whereby an ASR system identifies phonemes in speech, and a NMF algorithm is implemented using one of 40 component matrices, one for each phoneme. This model is somewhat limited in its application to babble, in the author's opinion, as the initial ASR algorithm's performance is likely to be low on a noisy signal. However, the algorithm did show that NMF performance is be improved by forming bases on phonemes.

A novel alternative model was recently proposed by Mohammadiha and Leijon [14], who used HMMs to form the basis of the NMF algorithm separating speech from noise , and thus used a temporally probabilistic method to form and estimate of clean speech. By doing so, the algorithm accounts for an extra feature outlined in Figure 2-2.

2.7 Summary

The following three paragraphs outline the three proposed focuses for this research.

Much work has been done in literature on speech enhancement under noisy conditions. Systems are available to improve intelligibility in telecommunication systems, or to improve the quality of ASR systems for home technology. However, most of the technologies outlined in this literature review have not been implemented in end-user systems, such as hearing aids or mobile technology. This is due to practicality constraints within the algorithms, in particular the training requirement, generally required to be specific to the end-user.

Additionally, most of the algorithms are tested for human recognition improvements, but rarely for improvements in machine recognition, highlighted in Figure 2-3. Little literature exists on the comparison of performance between these two forms of recognition. An area of research proposed is to investigate the relationship between human and machine recognition improvement under speech enhancement algorithms, to see if one implies the other.

Finally, improvements have been made in focusing recognition by forcing component matrices to represent phonemes. However, current algorithms require complicated implementations. It is proposed to attempt to simplify this process.

Chapter 3: Methodology

This chapter describes the method by which the research questions described in §1.1 will be addressed. Outlined are:

- the proposed algorithms;
- the infrastructure, both software and hardware, required to implement these algorithms;
- the data on which the algorithms shall be implemented for testing;
- and finally the evaluation measures by which the algorithms shall be assessed on.

3.1 Implementation Infrastructure

The algorithms outlined throughout the following section will be implemented in MATLAB. This environment was selected due to its availability and ease-of-use. MATLAB includes a number of tools useful in signal processing and statistical analysis [42, 43], and other authors have provided tools in the MATLAB language [39, 44, 45], all of which may be implemented within this thesis. A limitation of the MATLAB environment in signal processing is the comparative performance with lower-level languages.

Simulations are to be run on a MacBook Pro 64-bit system with an Intel Core i7 2.3GHz 4-core processor with 16GB memory, running OSX 10.9.2 and MATLAB R2013a.

3.2 Algorithms

A number of algorithms are to be used within this thesis. Firstly, an existing algorithm is to be implemented, as outlined in §3.2.1. The purpose of this is to re-evaluate the effectiveness of the algorithm in the context of machine recognition. This will also give a benchmark for proposed changes to be evaluated against. In §3.2.2 changes to the algorithm are proposed in order to force phoneme-dependant recognition. It is hypothesised this will force recognition to be improved and will allow improvements to be made in order to reduce the training data required as outlined in §3.2.3. Each algorithm will be tested and evaluated using the evaluations measures outlined in §3.4.

3.2.1. Algorithm I – Non-negative Matrix Factorisation with Priors

The NMF with priors algorithm is outlined by Wilson, et al. [15]. In this paper, the algorithm is evaluated using the segmental SNR and PESQ methods, both models for human perceptual quality. This algorithm shall be implemented in order to test the performance on machine understanding using an ASR system, as outlined in §3.4.

3.2.2. Algorithm II – Non-negative Matrix Factorisation of Phonemes with Priors

In order to improve the previous algorithm, adaptations are proposed to develop a phoneme oriented method. It was assumed that speech can be expressed as a sequence of phonemes, and that phonemes are constant throughout their duration, such that a single speaker speech signal may be expressed as:

$$S = \{p(1), p(2), \dots, p(T)\} \quad \textbf{(3-1)}$$

where S is the short time Fourier series sequence of length T and $p(t)$ is a single-frame Short Time Fourier Transform (STFT) of a single phoneme. Speech within babble noise will be of the form

$$X(t) = S_1(t) + S_2(t) + \cdots + S_N(t) \quad \textbf{(3-2)}$$

where $S_1(t)$ is the SoI and other speakers $S(t)$ are competing speakers. Therefore, it can be assumed that for any given t there are up to N different phonemes present.

The algorithm proposed, NMF of phonemes with priors, is a variation on that used by Wilson, et al. [15], with a novel adaptation to focus recognition on phonemes:

1. Calculate component matrices V_{Speech} and V_{Noise} by drawing phonemes from the SoI and from a range of speakers respectively.
    a. V_{Speech}: the number of samples of each phoneme drawn from the SoI to produce the spectral matrix will be varied from 1 to 1000.
    b. V_{Noise}: the number of samples of each phoneme drawn per speaker to produce the spectral matrix will be varied from 1 to 1000.
    c. A further component matrix, V_{all}, shall then calculated by concatenating V_{speech} and V_{noise}.
2. Calculate W_{speech} and W_{noise} by iterating update rule (2-10), and calculate the respective covariance of the log values of these matrices as Λ_{speech} and Λ_{noise}.
3. A variation on the multiplicative update rule (2-10) proposed by Wilson, et al. [15] will be used to calculate W_{all}:

$$W_{all(c,s)} \leftarrow W_{all(c,s)} \frac{\sum_f V_{all(f,c)} \dfrac{X_{f,s}}{(V_{all} W_{all})_{f,s}}}{\left[\sum_k V_{all(k,c)} + \alpha\varphi(W_{all})\right]_\epsilon} \quad \textbf{(3-3)}$$

where $[.]_\epsilon$ is an operator that rounds any zero or negative elements up to a small constant, ϵ, in order to enforce the non-negativity constraint; $\varphi(W_{all})$ is a function with the objective of enforcing that the Gaussian distribution of W_{all} be similar to that of W_{speech} and W_{noise}; α is a constant controlling the amount by which $\varphi(W_{all})$ will affect the results; and Λ_{all} is a matrix $\begin{bmatrix} \Lambda_{speech} & 0 \\ 0 & \Lambda_{noise} \end{bmatrix}$ representing the empirical covariances of the log values of W.

4. An estimate of clean speech is formed by:

$$\hat{X}_{speech} = V_{speech} W_{all\,1:n_c} \quad \textbf{(3-4)}$$

This algorithm is relatively simple in implementation, and provides separation by recognition of the SoI's voice. The motivation for using phonemes as bases is that speech signals are naturally phoneme driven. It is hypothesised that using phonemes as bases will result in separation more discriminative of the speaker's voice. Raj, et al. [26] demonstrated an increased separation of speech from music when using phoneme type separation, however the algorithm used was more complex, involving ASR systems to segregate phonemes. Here the segregation of phonemes is done as part of the NMF algorithm.

The use of drawn phonemes for the component matrices also allows different training mechanisms to be implemented. In the following section, novel and practical methods are proposed.

3.2.3. Reducing the Training Requirements

An issue with speech enhancement systems operating on characteristics of the SoI's voice is that they require *a priori* knowledge of the SoI's voice. The length of recordings required is generally quite large: Raj, et al. [26] used approximately 26 minutes of phoneme data, and Mohammadiha and Leijon [14] used 600 sentences. This becomes an issue in practical systems: in some applications this may be acceptable, but for most, especially for systems for mass distribution to the public, this in not acceptable.

The system outlined above requires 1 to 1000 bases per phoneme. At 40 phonemes and 20-40 ms per frame, this corresponds to over half an hour required for 1000 bases, or just 3 minutes if 100 bases are found to be adequate.

In order to gather the required test data in a practical system, for example for a mobile phone application or a smart hearing aid, two systems are proposed. The first requires the speaker to read a known set of sentences, i.e., the speaker will read a known passage. The second system operates on unknown passages, such that it could be incorporated into a mobile phone to train whilst the speaker has a natural conversation over the phone. Both systems require an ASR to recognise the speech at the phoneme level.

Prompted sentences shall be selected in order to provide as much phoneme variation as possible. Some phonemes are naturally more common in occurrence than others, and it is expected that some phonemes, such as /ZH/, will be more difficult to obtain. For these phonemes, the number of bases is allowed to be smaller.

Regardless of whether the free-speech or prompted method is used, the recording shall then be passed through an ASR algorithm for phoneme-level recognition [46]. The phoneme set to be used is the CMUDICT [47], based on the ARPAbet.

Once phonemes have been identified and synchronised with speech, samples can then be drawn. Samples are to be drawn randomly from within the bounds of the phoneme, i.e., not all from the beginning or the end of the phoneme. This is to prevent bias, and to ensure a strong model. The actual number of samples drawn is subject to the test parameters and will be varied. The drawn samples are then concatenated to form the spectral component matrix.

A further option, to be implemented and tested, is to further apply NMF training, using the above formed spectral component matrix as an input. Multiplicative update rules will be used, as they are more sensitive to initial conditions than ALS algorithms [31].

3.3 Test Data

Test data to be used shall be a mixture of speakers from the Wall Street Journal (WSJ) corpus [48]. Different recordings will be reserved for testing and training. Tests signals will be a speech signal for the SoI added to a number of other signals representing the babble noise.

Tests will begin under conditions defined as lower difficulty, and the difficulty shall be until increased the algorithms perform no significant increase in recognition. Definitions of difficulty are given in Table 3-1.

Table 3-1 – Difficulty Classification

	Lower Difficulty	**Higher Difficulty**
Gender	Competing speaker(s) different gender to SoI	Competing speakers(s) same gender as SoI
Number of competing speakers	2	More than 2
Relative level of SoI to competing speakers	10dB	0dB

The first test will be conducted on a mixture of two speakers, of different genders, with the competing speaker added at a level of -10 dB relative to the SoI. Subsequent test will be conducted with mixtures of a higher difficulty.

Prior to mixing, all signals shall be downsampled to 16 kHz and trimmed to a common length. Competing Speaker signals shall be adjusted to the appropriate mixture levels by comparing the average power level with that of the SoI signal average power level, and mixed in the time domain to form the mixture signal.

The STFT shall be calculated with 20-40 ms frames, giving 320-640 samples and therefore 161-321 frequency bins. The exact values shall be empirically determined to balance shorter time frames giving higher temporal resolution and longer time frames giving higher spectral resolution. Frames will have approximately 50 % overlap.

3.4 Evaluation Measures

An objective of this thesis is to develop a system applicable to a range of practical applications. Proposed algorithms, therefore, must be evaluated on their ability to improve recognition both for human listeners and machine listener, as outlined in research question 1. Three evaluation measures are to be used in order to achieve this.

The first to be used in the ITU standard MOS [18]. This measure has been selected to evaluate true human perceptual evaluations. Due to limited resource, it is likely such tests will be required to be conducted on a relatively small test base.

The second evaluation measure to be used in PESQ [19]. This is another measure for human perceptual quality. This method provides ease of implementation, through use of a MATLAB program provided by Loizou [45], ease of testing which can be automated, and comparability with results of other papers, which often use the PESQ evaluation algorithm.

The final test measure will be recognition accuracy through an ASR system. This method is to be used to evaluate machine recognition. The ASR system to be used is that distributed with the CHiME challenge, such that results may be compared with the results of the entries.

Data gathered through the methods proposed here should be sufficient to answer the proposed research questions.

3.5 Preliminary Progress

A basic system has been implemented in MATLAB to produce V_{Speech} using the drawn phoneme method. The code and results are attached as Appendix C: Component Matrix using Drawn Phonemes.

3.6 Project Management

See attached Appendix B: Work Breakdown Structure.

3.7 Resource Management and Cost Analysis

A budget of $250 has been allocated towards this research.

Resources required for research are:

- MATLAB Student licence
- Computer hardware
- WSJ Corpus

All required resources are available under existing infrastructure, thus it is not expected any further funding is required.

References

[1] E. C. Cherry, "Some Experiments on the Recognition of Speech, with One and with Two Ears," *The Journal of the Acoustical Society of America,* vol. 25, pp. 975-979, 1953.

[2] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature,* vol. 401, pp. 788-791, 1999.

[3] M. Cooke, J. R. Hershey, and S. J. Rennie, "Monaural speech separation and recognition challenge," *Computer Speech & Language,* vol. 24, pp. 1-15, 2010.

[4] J. Barker, E. Vincent, N. Ma, H. Christensen, and P. Green, "The PASCAL CHiME speech separation and recognition challenge," *Computer Speech & Language,* vol. 27, pp. 621-633, 2013.

[5] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second 'CHiME'speech separation and recognition challenge: Datasets, tasks and baselines," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 126-130.

[6] N. Krishnamurthy and J. H. Hansen, "Babble noise: modeling, analysis, and applications," *Audio, Speech, and Language Processing, IEEE Transactions on,* vol. 17, pp. 1394-1407, 2009.

[7] P. C. Loizou, "Noise Compenation by Human Listeners," in *Speech Enhancement: Theory and Practice, Second Edition*, ed: Taylor & Francis, 2013.

[8] D. Wang, "Time—Frequency Masking for Speech Separation and Its Potential for Hearing Aid Design," *Trends in amplification,* vol. 12, pp. 332-353, 2008.

[9] L.-P. Yang and Q.-J. Fu, "Spectral subtraction-based speech enhancement for cochlear implant patients in background noise," *The Journal of the Acoustical Society of America,* vol. 117, pp. 1001-1004, 2005.

[10] J. Benesty, S. Makino, and J. Chen, *Speech enhancement*: Springer, 2005.

[11] I. Pollack and J. M. Pickett, "Stereophonic Listening and Speech Intelligibility against Voice Babble," *The Journal of the Acoustical Society of America,* vol. 30, pp. 131-133, 1958.

[12] P. C. Loizou, "Classes of Speech Enhancement Algorithms," in *Speech Enhancement: Theory and Practice, Second Edition*, ed: Taylor & Francis, 2013.

[13] S. Kamath and P. Loizou, "A multi-band spectral subtraction method for enhancing speech corrupted by colored noise," in *IEEE international conference on acoustics speech and signal processing*, 2002, pp. 4164-4164.

[14] N. Mohammadiha and A. Leijon, "Nonnegative HMM for babble noise derived from speech HMM: Application to speech enhancement," 2013.

[15] K. W. Wilson, B. Raj, P. Smaragdis, and A. Divakaran, "Speech denoising using nonnegative matrix factorization with priors," in *ICASSP*, 2008, pp. 4029-4032.

[16] S. Dupont and J. Luettin, "Audio-visual speech modeling for continuous speech recognition," *Multimedia, IEEE Transactions on,* vol. 2, pp. 141-151, 2000.

[17] N. Mohammadiha, W. B. Kleijn, and A. Leijon, "Gamma Hidden Markov Model as a Probabilistic Nonnegative Matrix Factorization," 2013.

[18] International Telecommunication Union, "P.800: Methods for subjective determination of transmission quality," *ITU-T Recommendation,* 1996.

[19] International Telecommunication Union, "P.862: Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs," *ITU-T Recommendation,* p. 862, 2001.

[20] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, 2001, pp. 749-752.

[21] A. W. Rix, "Comparison between subjective listening quality and P. 862 PESQ score," *Proc. Measurement of Speech and Audio Quality in Networks (MESAQIN'03), Prague, Czech Republic,* 2003.

[22] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement," *Audio, Speech, and Language Processing, IEEE Transactions on,* vol. 16, pp. 229-238, 2008.

[23] M. Schmidt and R. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," 2006.

[24] B. Raj, R. Singh, and P. Smaragdis, "Recognizing speech from simultaneous speakers," in *INTERSPEECH*, 2005, pp. 3317-3320.

[25] B. Raj and P. Smaragdis, "Latent variable decomposition of spectrograms for single channel speaker separation," in *Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on*, 2005, pp. 17-20.

[26] B. Raj, R. Singh, and T. Virtanen, "Phoneme-Dependent NMF for Speech Enhancement in Monaural Mixtures," in *INTERSPEECH*, 2011, pp. 1217-1220.

[27] C. Févotte, "Majorization-minimization algorithm for smooth Itakura-Saito nonnegative matrix factorization," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, 2011, pp. 1980-1983.

[28] D. L. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?," in *NIPS*, 2003.

[29] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics,* vol. 5, pp. 111-126, 1994.

[30] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2000, pp. 556-562.

[31] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational Statistics & Data Analysis,* vol. 52, pp. 155-173, 2007.

[32] L. Finesso and P. Spreij, "Nonnegative matrix factorization and I-divergence alternating minimization," *Linear Algebra and its Applications,* vol. 416, pp. 270-287, 2006.

[33] L. Finesso and P. Spreij, "Approximate nonnegative matrix factorization via alternating minimization," *arXiv preprint math/0402229,* 2004.

[34] R. Albright, J. Cox, D. Duling, A. N. Langville, and C. Meyer, "Algorithms, initializations, and convergence for the nonnegative matrix factorization," North Carolina State University, 2006.

[35] S. Wright and J. Nocedal, "Derivative-Free Optimization," in *Numerical optimization*. vol. 2, ed: Springer New York, 1999, pp. 229-233.

[36] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural computation,* vol. 19, pp. 2756-2779, 2007.

[37] M. H. Van Benthem and M. R. Keenan, "Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems," *Journal of chemometrics,* vol. 18, pp. 441-450, 2004.

[38] R. Bro and S. De Jong, "A fast non-negativity-constrained least squares algorithm," *Journal of chemometrics,* vol. 11, pp. 393-401, 1997.

[39] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *The Journal of Machine Learning Research,* vol. 5, pp. 1457-1469, 2004.

[40] P. O. Hoyer, "Non-negative sparse coding," in *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, 2002, pp. 557-565.

[41] A. Cichocki, R. Zdunek, and S.-i. Amari, "New algorithms for non-negative matrix factorization in applications to blind source separation," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, 2006, pp. V-V.

[42] T. P. Krauss, L. Shure, and J. Little, *Signal processing toolbox for use with MATLAB®: user's guide*: The MathWorks, 1994.
[43] B. Jones, *MATLAB: Statistics Toolbox; User's Guide*: MathWorks, 1997.
[44] M. Brookes, "Voicebox: Speech processing toolbox for matlab," *Software, available [Mar. 2011] from www. ee. ic. ac. uk/hp/staff/dmb/voicebox/voicebox. html,* 1997.
[45] P. C. Loizou, "MATLAB PESQ Implementation," in *"Speech enhancement: Theory and Practice"*, ed: CRC Press, 2008.
[46] K. Gorman. (2007, 5/5/14). *Automatic Speech Segmentation with HTK* [Tutorial]. Available: http://www.ling.upenn.edu/~kgorman/speechseg.html/.speechseg.html
[47] R. Weide, "The Carnegie mellon pronouncing dictionary [CMUDICT. 0.6]," *Carnegie MellonUniversity: http://www.speech.cs.cmu.edu/cgi-bin/cmudict.,* vol. 9, 2005.
[48] T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals, "WSJCAMO: A British English speech corpus for large vocabulary continuous speech recognition," in *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, 1995, pp. 81-84.

Appendix A: Risk Assessment

Appendix B: Work Breakdown Structure