Speech Separation using Non-negative Features and Sparse Non-negative Matrix Factorization

Mikkel N. Schmidt

Technical University of Denmark
Richard Petersens Plads, Bldg. 321
DK-2800 Kgs. Lyngby, Denmark

Abstract

This paper describes a method for separating two speakers in a single channel recording. The separation is performed in a low dimensional feature space optimized to represent speech. For each speaker, an overcomplete basis is estimated using sparse non-negative matrix factorization, and a mixture is separated by mapping the mixture onto the joint bases of the two speakers. The method is evaluated in terms of word recognition rate on the speech separation challenge data set.

Key words: Speech separation challenge, Sparse non-negative matrix factorization

1 Introduction

It is not known how the human auditory system is able to separate sound sources, but its ability to do so is remarkable. Yet, no machine has been constructed that in general can separate sounds as well as humans can, but due to the continuing joint effort of signal processing scientists and psychologists the gap between human and computer performance is closing.

Humans are able to use information at different levels to accomplish the separation task. When we listen with our two ears, we use spatial information to separate sources based on their location. Scientists have been fairly succesful in using the information availably in multiple channels to separate sources, either based on spatial location using beamforming techniques or based on statistical independence between sources using independent component analysis techniques. But perhaps more important, humans use spectral and dynamic

Email address: mns@imm.dtu.dk (Mikkel N. Schmidt).

characteristics of the sources to perform separation. Even when listening with only one ear, we can effortlessly separate sound sources. The exact mechanisms regulating this ability are not yet fully understood. In addition to this, humans also use aquired high-level knowledge about sound sources to aid the segregation.

In this paper I present a data-driven approach to speech separation based on spectral characteristics of the speakers without modeling the temporal structure. I compute a spectral basis for each speaker using sparse non-negative matrix factorization (NMF). Given a mixture, I estimate which speakers are present and use the learned bases to separate them. The method is evaluated on the speech separation challenge test set, which is introduced in the next section.

1.1 The Speech Separation Challenge

At the Interspeech 2007 conference in Pittsburgh the first large scale comparison of methods for separating and recognizing speech was conducted: the speech separation challenge. The rules of the challenge are available online [1] along with papers from several authors describing results so far.

The training and test data are taken from the GRID corpus (Cooke and Shao, 2006) which consists structured sentences with a small vocabulary illustrated in Figure 1. The corpus has 34 speakers and has a total of 34.000 spoken sentences, half of which are used as training data in the challenge. The test set is constructed by mixing two different sentences at different target-to-masker ratios (TMR) — one of which, the target, always says the word *white*. In addition to this there is a test set where the interference is speech shaped noise. The task is to recognize the letter and the number spoken by the target speaker.

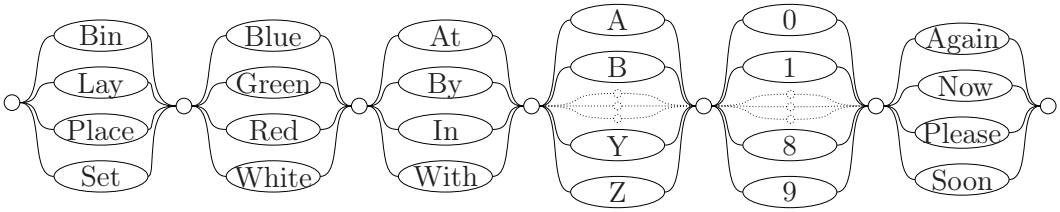

Fig. 1. Sentence structure of the GRID corpus.

[1] `http://www.dcs.shef.ac.uk/~martin/SpeechSeparationChallenge.htm`

2 Speech Separation using Sparse NMF

The main idea in this paper is to use sparse non-negative matrix factorization as a means of separating speech. The method relies on the speaker's having different spectral characteristics. A block diagram outlining the framework is shown in Figure 2.

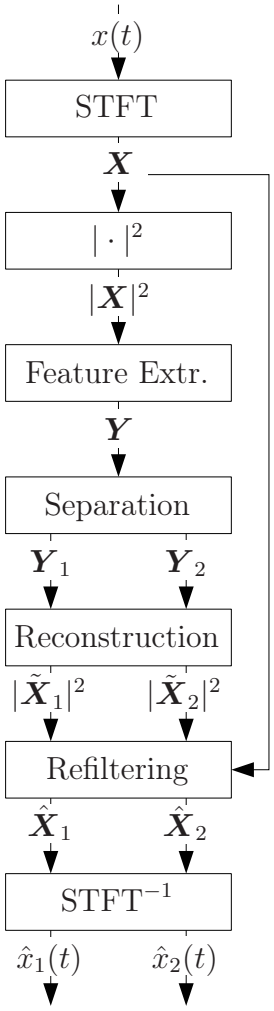

Fig. 2. Block diagram of the speech separation framework.

I transform the mixed signal to the time-frequency domain using the short time Fourier tranform (STFT) and take the magnitude squared to get the power spectrogram. I then reduce the dimensionality of the power spectrogram by mapping it onto a lower dimensional basis that is learned from training data using non-negative matrix factorization (NMF). This has the advantage of reducing the required computations in the following steps, but, equally important, it serves to emphasize the most important frequency regions, since the features are optimized to represent speech.

For each speaker in the training set, I estimate an overcomplete basis in the feature space using sparse NMF. This basis can be seen as a speaker dependent non-parametric generative model, i.e., the observations for a specific speaker are generated as non-negative linear combinations of elements in this basis. Putting it differently, observations lie in the subspace of the convex cone spanned by the basis vectors (Donoho and Stodden, 2003).

To separate mixed speech I map the observed features on to the concateneted bases of the two speakers in the mixture. Separation is then performed by reconstructing the parts pertaining to each speaker individually. This gives an estimate of each speaker in the feature space, which I then map back to the power spectrogram space. Here, I compute smoothed time varying Wiener filters which are used to filter the original mixture giving the final result.

In the following, I discuss each of these steps in further detail, begining with a review of sparse NMF.

2.1 Sparse Non-negative Matrix Factorization

Non-negative matrix factorization (Lee and Seung, 1999) is a method by which a matrix, $\boldsymbol{X}$, is approximated by the product of two matrices, $\boldsymbol{W}$ and $\boldsymbol{H}$, enforcing the constraint that all matrices are non-negative,

$$\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H} \quad \text{s.t. } \boldsymbol{W}, \boldsymbol{H} \geq 0. \tag{1}$$

Since no elements are allowed to be negative and, thus, all combinations are additive, the factorization often leads to a *parts based* representation (Lee and Seung, 1999). The parts I hope to find here are basic feature vectors, that are specific and representative for a speaker. There exists a number of algorithms for computing such a factorization (Lee and Seung, 2000; Lin, 2007; Kim et al., 2007).

Non-negative matrix factorization should perhaps more accuratly be called non-negative matrix *approximation*, since most algorithms seek to minimize some divergence measure, D, between the data matrix and the approximating factorization (Dhillon and Sra, 2005),

$$\min_{\boldsymbol{W},\boldsymbol{H}\geq 0} D(\boldsymbol{X}, \boldsymbol{W}\boldsymbol{H}). \tag{2}$$

A number of different divergence functions have been suggested (Cichocki et al., 2006; Dhillon and Sra, 2005; Kompass, 2007) corresponding to different assumptions about the error of the approximation. The least squares cost function, $D_{\text{LS}}(\boldsymbol{A}, \boldsymbol{B}) = \sum_{i,j}(\boldsymbol{A}_{i,j} - \boldsymbol{B}_{i,j})^2$, for example, corresponds to a Gaussian error model whereas the generalized Kullback-Leibler (KL) divergence,

$D_{\mathrm{KL}}(\boldsymbol{A}, \boldsymbol{B}) = \sum_{i,j} \boldsymbol{A}_{i,j} \log \frac{\boldsymbol{A}_{i,j}}{\boldsymbol{B}_{i,j}} - \boldsymbol{A}_{i,j} + \boldsymbol{B}_{i,j}$, effectively corresponds to a Poission process (Sajda, 2003).

I believe that the KL divergence is a reasonable cost function when using NMF to approximate a power spectrogram of speech. In a Poisson process the variance is equal to the mean, which implies that the error of the approximation at a certain frequency bin will be proportionate to the magnitude of that bin. This is reasonable, since humans percieve noise in proportion to the signal of interest. Effectively, compared to the squared error measure, this gives more weight to the high frequency components, which typically are small in magnitude compared with the low frequency components.

Sparse NMF (Hoyer, 2002) is an extension of NMF, in which an additional sparsity constraint is enforced on the matrix $\boldsymbol{H}$, i.e., a solution is sought where only a few basis vectors are active simultaneously. The sparse NMF problem can be formulated (Dhillon and Sra, 2005) as

$$\min_{\boldsymbol{W},\boldsymbol{H} \geq 0} D(\boldsymbol{X}, \boldsymbol{W}\boldsymbol{H}) + \beta(\boldsymbol{H}), \tag{3}$$

where β is a penalty term that enforces the sparsity. This penalty could be selected as the 0-norm, i.e., the count of non-zero elements in $\boldsymbol{H}$, but this leads to a very rough cost function that is hard to minimize because of its many local minima. A penalty function that leads to a smoother regularization (Hoyer, 2002) while still inducing sparsity is the the 1-norm, which, in Bayesian terms, corresponds to assuming an exponential prior over $\boldsymbol{H}$. In practice I use $\beta(\boldsymbol{H}) = \lambda \sum_{i,j} \boldsymbol{H}_{i,j}$, where λ is a parameter which controls the tradeoff between sparsity and accuracy of the approximation. To use this penalty function I must also introduce a normalization constraint on either $\boldsymbol{W}$ or $\boldsymbol{H}$, since trivial solutions minimizing β can be found by letting $\boldsymbol{H}$ decrease and $\boldsymbol{W}$ increase accordingly. Here, I choose to normalize the basis, i.e., the columns of the matrix $\boldsymbol{W}$.

With the sparseness penalty, the data is modeled not only as a non-negative linear combination of a set of basis vectors, but as linear combinations using only a few basis vectors at a time. This allows me to compute an overcomplete factorization, i.e., a factorization with more basis vectors than the dimensionality of the data. Without the sparsity constraint, any basis spanning the entire positive orthant would be a solution. For example, the unit basis is a complete (uninteresting) solution to all NMF problems: $\boldsymbol{X} = \boldsymbol{I}\boldsymbol{X}$. With the sparsity constraint, however, the solution will be a set of basis vectors that lie close to the data points and contain all data points inside the convex cone which they span.

In my experiments I used the following simple multiplicative update rules (Lee and Seung, 2000; Eggert and Körner, 2004) to minimize Equation (3). The

algorithm starts with randomly initialized matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ and alternates the following updates until convergence,

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \bullet \frac{\bar{\boldsymbol{W}}^\top \frac{\boldsymbol{X}}{\bar{\boldsymbol{W}}\boldsymbol{H}}}{\bar{\boldsymbol{W}}^\top \boldsymbol{1} + \lambda}, \tag{4}$$

$$\boldsymbol{W} \leftarrow \bar{\boldsymbol{W}} \bullet \frac{\frac{\boldsymbol{X}}{\bar{\boldsymbol{W}}\boldsymbol{H}}\boldsymbol{H}^\top + \bar{\boldsymbol{W}} \bullet \left(\boldsymbol{1}\left(\boldsymbol{1}\boldsymbol{H}^\top \bullet \bar{\boldsymbol{W}}\right)\right)}{\boldsymbol{1}\boldsymbol{H}^\top + \bar{\boldsymbol{W}} \bullet \left(\boldsymbol{1}\left(\frac{\boldsymbol{X}}{\bar{\boldsymbol{W}}\boldsymbol{H}}\boldsymbol{H}^\top \bullet \bar{\boldsymbol{W}}\right)\right)}. \tag{5}$$

In the equation, $\bar{\boldsymbol{W}}$ is the columnwise normalized basis matrix; $\boldsymbol{1}$ is a square matrix of suitable size with all elements equal to 1; the bold operators indicate pointwise multiplication and division; and λ is the regularization parameter used to adjust the level of sparsity.

2.2 Non-negative Features

When modeling audio spectra it is sensible to use a representation in which the human perception of both amplitude and frequency is taken into account — a representation such as a magnitude compressed, Mel frequency spectrogram. In this work I take a different route to the same goal. As I have already discussed, the KL divergence serves to account for the relative perception of amplitude in the human auditory system. Now, in stead of computing spectral vectors on a perceptually motivated frequency scale, I compute linear frequency spectral vectors and map them onto a set of basis vectors which are optimized to encode speech. Mapping the power spectrogram into this feature space serves both to emphasize the important frequencies and to reduce the dimensionality of the data.

I concatenate 10 minutes of speech from the 34 different speakers in the training set and compute the magnitude squared STFT, $|\boldsymbol{X}_F|^2$ (ideally I should use the entire training set, but I use only a subset to save computation.) I use a block size of 20 ms and 50 percent overlap, which gives me a 257 dimensional spectral representation. I then compute a lower dimensional basis using the NMF updates in Equation (4–5),

$$|\boldsymbol{X}_F|^2 \approx \boldsymbol{W}_F \boldsymbol{Y}_F \tag{6}$$

which gives me the feature basis matrix $\boldsymbol{W}_F$. The sparsity parameter, λ, is set to zero in the computation since there is no need for sparsity when using NMF to reduce the dimensionality of the data. Then, to map a power spectrogram,

$|\boldsymbol{X}|^2$, into the feature space I use Equation (4) to compute $\boldsymbol{Y}$,

$$|\boldsymbol{X}|^2 \approx \boldsymbol{W}_F \boldsymbol{Y}. \tag{7}$$

Mapping back to the power spectrogram from the feature space is simply done by pre-multiplying the feature matrix with $\boldsymbol{W}_F$.

The feature basis, $\boldsymbol{W}_F$, is shown in Figure 3 for a 32-dimensional decomposition. The basis appears to have almost constant center frequency-to-bandwidth ratio as seen when plotted on a logarithmic frequency axis, although it has slightly more resolution in the mid frequency range. I find it very appealing that, in line with the findings of Lewicki (2002), optimal encoding leads to a representation which very much resembles that of the peripheral auditory system.

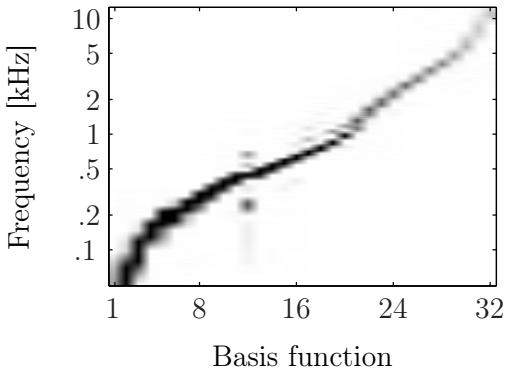

Fig. 3. Basis, $\boldsymbol{W}_F$, of the 32-dimensional feature space computed by non-negative matrix factorization. To aid the visualization, the basis vectors are manually sorted according to center frequency.

I chose the number of features empirically by listening to the quality of the optimal separation attainable at different feature dimensionalities. For a few different combinations of speakers I separated their mixtures using the optimal time varying Wiener filter based on knowledge of the power spectrum of the clean speech. When working directly in the power spectrogram, the optimal time varying Wiener filter yields near perfect separation, even in the difficult case of separating two sentences spoken by the same speaker. When I listened to the result obtained working in different dimensionality-reduced feature spaces, I found that even as little as 16 features gave very good results, and when using 32 features the separation was indistinguishable from separation performed in the power spectrogram domain.

Given the feature space representation, $\boldsymbol{Y}$, of a mixture of two speakers, I now consider how to perform the separation. For the moment I assume that the speakers are known—I will discuss how to estimate their identity in the following section.

The first step is to compute an overcomplete basis for each speaker. I concatenate 2.5 minutes of training speech from each speaker and map it into the feature space as described in the previous section (ideally, again, I should use the entire training set; I use a subset to save computation.) Then I compute an overcomplete sparse NMF decomposition,

$$\boldsymbol{Y}_n^{\text{train}} \approx \boldsymbol{W}_n^{\text{train}} \boldsymbol{H}_n^{\text{train}}, \tag{8}$$

using Equation (4–5). I empirically choose the sparsity parameter $\lambda = 0.5$. This gives me a basis matrix, $\boldsymbol{W}_n^{\text{train}}$, for each speaker, examples of which are shown in Figure 4.

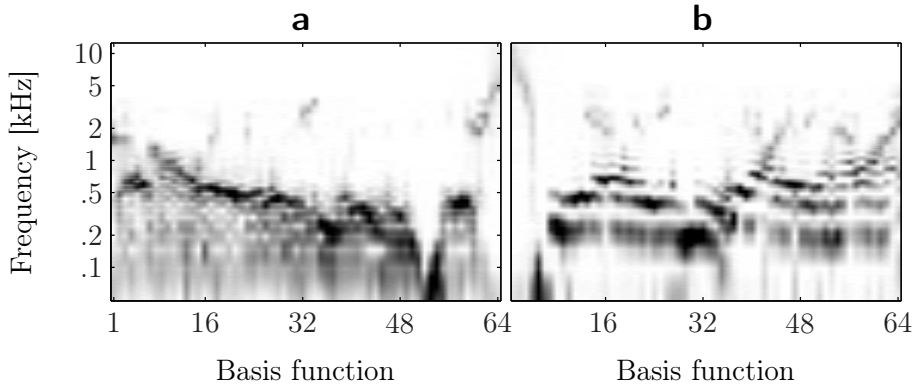

Fig. 4. Two times overcomplete bases for (**a**) a male and (**b**) a female speaker. For the visualization, the basis vectors are sorted to maximize neighbor correlation.

Next, I map the feature space representation of the mixture onto the joint bases of the speakers using Equation (4) to compute $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$,

$$\boldsymbol{Y} \approx \left[\boldsymbol{W}_1^{\text{train}}, \boldsymbol{W}_2^{\text{train}}\right] \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2 \end{bmatrix}. \tag{9}$$

Finally, the estimate of each speaker in the feature space is given as

$$\boldsymbol{Y}_n = \boldsymbol{W}_n^{\text{train}} \boldsymbol{H}_n. \tag{10}$$

I chose the number of basis vectors empirically by listening to separated speech waveforms for few different combinations of speakers. I experimented with

bases that were between 2 and 16 times overcomplete and found that, perceptually, the results did not vary much. In my evaluations on the challenge test set I used two times overcomplete bases.

I have illustrated the separation method graphically in Figure 5, only in three dimensions, though, since it is difficult to draw 32-dimensional vectors. The two sets of dots represent the training data used to compute the sparse overcomplete bases which are depicted by the two sets of arrows. As described earlier, due to the sparsity constraint, the basis vectors will lie inside or close to the cloud of training data, which, on the other hand, will lie inside the convex cone spanned by the basis vectors. A feature vector, $\triangle$, from a mixture of the two speakers lies outside the subspaces of both speakers, but can be described as the sum of a vector from each subspace, $\square$ and $\diamond$. The sparse NMF method separates the mixture by mapping a mixed feature vector onto the joint subspaces of the sources and then computing the part which falls in each subspace.

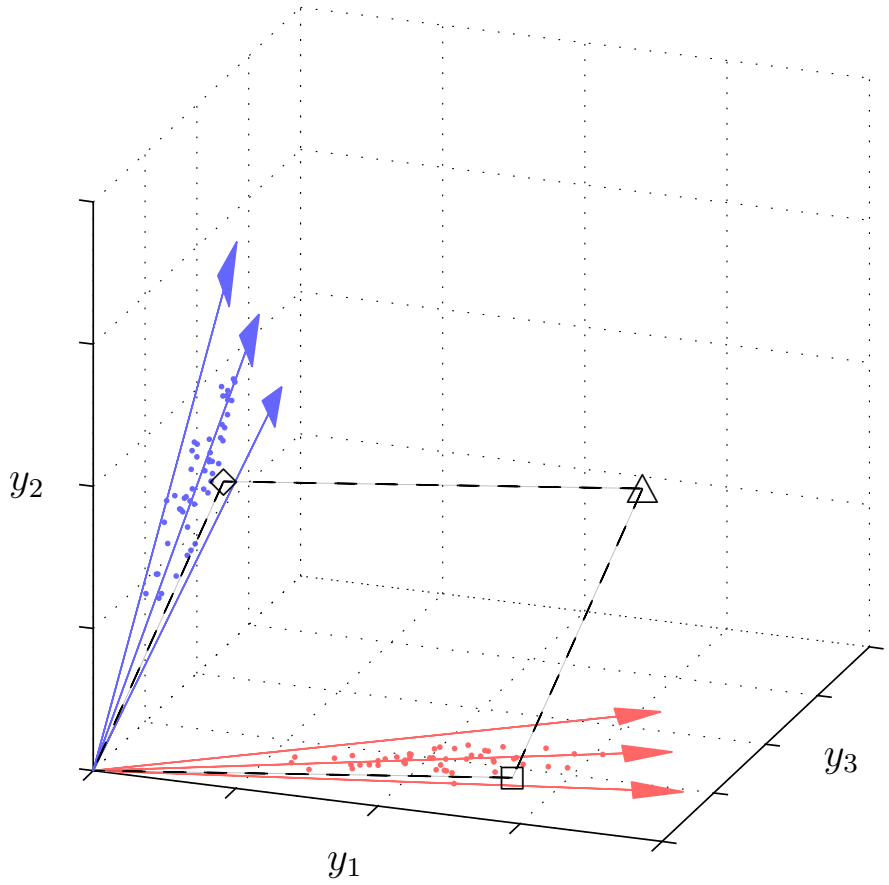

Fig. 5. Illustration of signal separation using sparse coding. See text for details.

The output of the separation algorithm is estimates of the separated speech

in the feature space, $\boldsymbol{Y}_n$, which I can map back into the power spectrogram representation, $|\tilde{\boldsymbol{X}}_n|^2 = \boldsymbol{W}_F \boldsymbol{Y}_n$. Now, all that is left is to compute separated audio waveforms. I discuss this in Section 2.5, but first I consider the estimation of speaker identity.

2.4 Speaker Recognition

An important part of the presented framework for speech separation is to efficiently estimate the identity of the speakers present in a mixture. This is, however, not the main focus of this paper—here, I just present a quick and easy approach akin to that of Kristjansson et al. (2006). I provide results based on this method as well as based on oracle knowledge of the speaker identity.

To estimate the speakers present in a mixture, I assume that in a mixed sentence each speaker's voice will be present in isolation in small glimpses whenever the other speaker pauses. I start by mapping the mixture into the non-negative feature domain. Here, I normalize the mixture, map it onto each of the 34 speakers' overcomplete bases, and compute the total error measure (the generalized KL divergence) for each time frame. Since none of the speaker models are a good match to a mixture of two speakers, the error will be relatively high except at frames where only one voice is present. For each speaker I compute the average of the 10 percent smallest errors, to get a measure of how well the model fits when it fits best. I then select the two speakers with the lowest error score.

On the speech separation challenge test set, this simple method identifies at least one of the two speakers in the mixture with 98 percent success, but it only finds the correct identity of both speakers 20 percent of the time. When the gender of the two speakers are the same, the method does find two speakers of that gender; however, in a mixture of a male and female speaker, the method in more than half the cases wrongly suggests two speakers of the same gender.

I am aware that, with more advanced methods, it is possible to estimate the identity of the speakers in the challenge test set almost perfectly, e.g., using combinatorial search (Schmidt and Olsson, 2006; Kristjansson et al., 2006), but it interesting to see if using a less accurate, and much less computationally expensive, speaker identification algorithm significantly worsens the performance of the speech separation algorithm. Could using two speaker models that are a good fit, but which do not correspond to the exact speakers in the mixture, be sufficient to separate the speech?

The output of the separation algorithm is an estimate of the power spectrogram for each speaker. One could speculate that this estimate could simply be combined with the phase information of the mixed signal to compute a waveform by the inverse STFT. While this is certainly possible, greater flexibility and much better perceptual sound quality can be obtained by using the estimated power spectrograms to *refilter* the mixture signal.

The idea of refiltering (Roweis, 2003; Srinivasan et al., 2006) is to separate the sound sources by filtering the mixed signal with a time-varying filter, $|\boldsymbol{B}_n|$, designed to preserve time-frequency regions containing only the signal of interest and attenuate regions containing the interfering signal: $\hat{\boldsymbol{X}}_n = \boldsymbol{X} \bullet |\boldsymbol{B}|_n$. Assuming that each time-frequency bin in a STFT representation is dominated by a single speaker, an often used filter is the binary time-frequency mask,

$$|\boldsymbol{B}_1^{\mathrm{BM}}|^2 = \begin{cases} 1, & |\tilde{\boldsymbol{X}}_1|^2 > |\tilde{\boldsymbol{X}}_2|^2 \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Another approach is the time varying Wiener filter where each time-freqency bin is multiplied by the estimated ratio of the target to the mixed signal,

$$|\boldsymbol{B}_1^{\mathrm{Wiener}}|^2 = \frac{|\tilde{\boldsymbol{X}}_1|^2}{|\tilde{\boldsymbol{X}}_1|^2 + |\tilde{\boldsymbol{X}}_2|^2}. \tag{12}$$

When the estimates of the target and interference are accurate, both methods provide very good sounding results, but when the estimates are less precise, artifacts are introduced in the signal. A common artifact known as musical noise occurs when narrow band components pop in and out of the estimated STFT spectrum. Another problem with these formulations is that multiplying by an arbitrary filter in the frequency domain corresponds to a circular convolution in the time domain with a non-causal filter, which leads to discontinuities between frames.

I propose a simple way to alleviate these problems based on the time varying Wiener filter. For each time frame I design a causal linear phase FIR filter to match the required frequency response using the frequency sampling method. Circular convolution is avoided by filtering the signal using overlap and add. To remove the musical noise artifact I smooth the filter coefficients by averaging over adjacent frames which reduces sudden changes in the filter from frame to frame. While removing the musical noise, however, this also allows more of the interfering speakers voice to remain in the resulting signal. I find that smoothing the time varying filter is an effective means to trading artifacts for

residual noise.

The resulting separated speech signals, when refiltered using this technique, have no audible artifacts, and the voice of the interfering speaker is significantly attenuated. An example of the speech separation method including the refiltering is shown in Figure 6. Notice, in the 0.25–0.3 seconds range of the female speaker, there is a visible error in the estimated power spectrum which is smoothed out in the refiltered power spectrum.

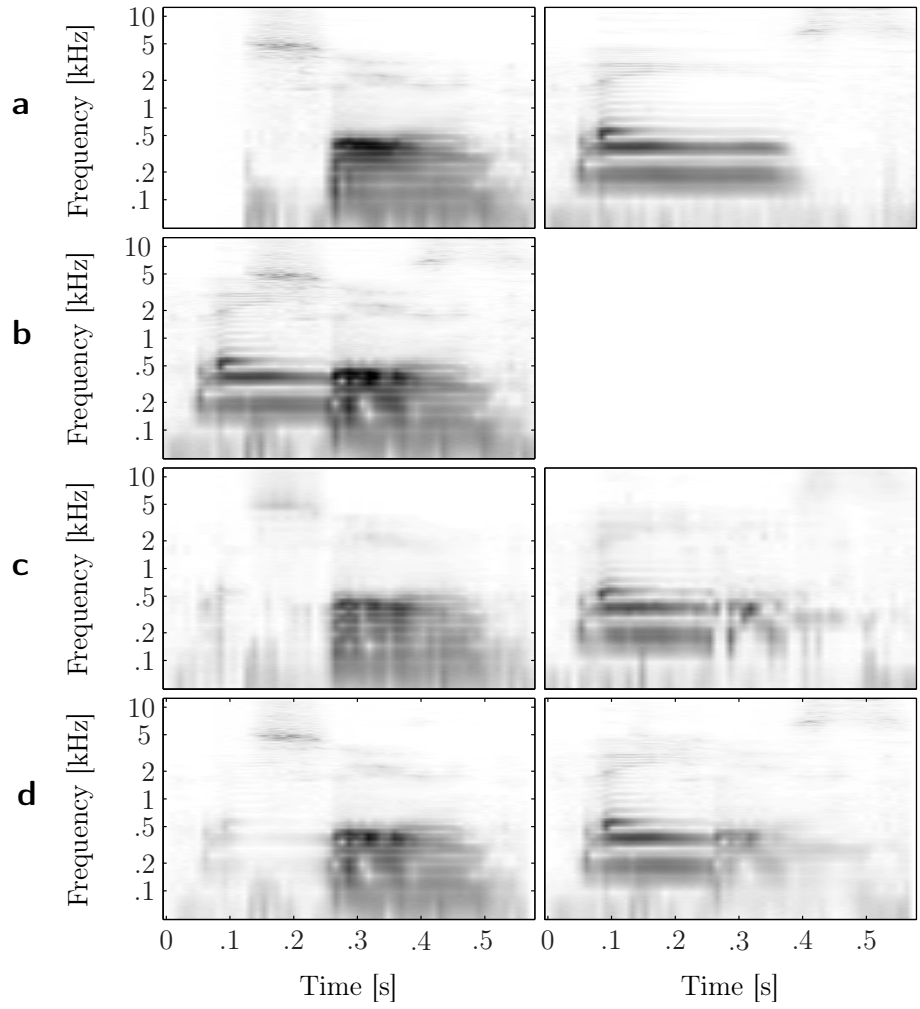

Fig. 6. Example of the speech separation method. (**a**) Left: male speaker saying the word *soon*. Right: female speaker saying the word *Please*. (**b**) Mixture of the two speakers. (**c**) Estimated spectrograms for the two speakers (**d**) Resulting spectrograms after smoothed time varying Winer filtering. This example is particularly difficult in the 0.25–0.4 seconds range, where the first formant of both speakers coincide around 400 Hz.

2.6 Speech Recognition

I evaluated the speech separation method on the speech separation challenge dataset described in section 1.1 using the reference HTK speech recognition system provided by the organizers of the challenge. The results are shown in Figure 7 and 8.

When the speaker identity is known beforehand, the sparse NMF improves the recognition rate significantly in all conditions except for the same speaker case. When using the speaker identification method, recognition rates are only slightly improved at low TMR but worsened at high TMR compared to the baseline (no speech separation.) For the speech shaped noise problem, the method does not improve the recognition rate.

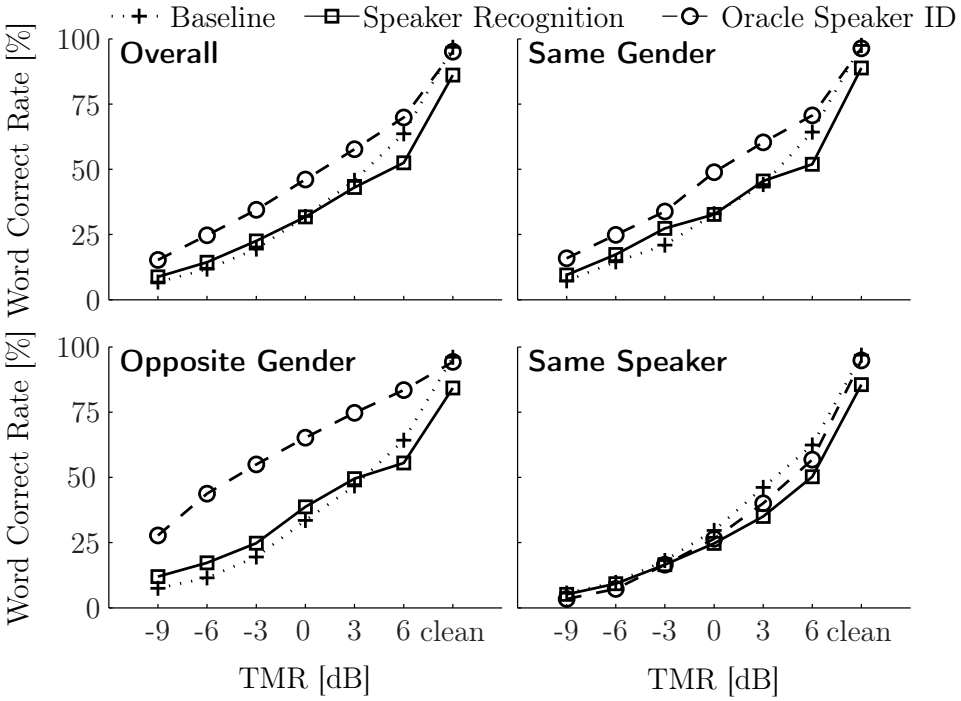

Fig. 7. Word recognition rates at different conditions for the two-talker problem.

3 Discussion

It appears that the correct identification of the speakers is crucial for the performance of the speaker dependent speech separation method presented: there is a very large difference between the recognition rates obtained when using knowledge of the speaker identity and when using the imperfect speaker identification method.

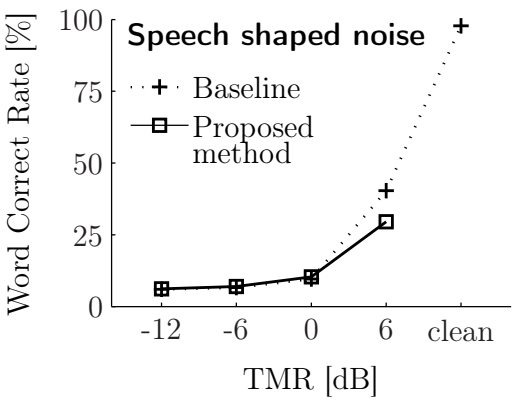

Fig. 8. Word recognition rate for the speech shaped noise problem.

If a speech separation system were to be used for a large number of unknown speakers, it would not be possible to train speaker dependent models beforehand. This, more general problem, is perhaps more important than developing sophisticated methods for separating a small set of known speakers. One way to use the ideas in this paper to solve such a problem could be to employ a battery of generic speaker models and use those who fit the signal at hand to perform the separation. My results, unfortunately, do not provide support for the feasibility of this idea. Certainly, more efficient methods for estimating speaker identity must be concieved for this idea to work.

An advantage of the sparse NMF approach to speech separation is its simplicity. It requires no grammatical model; in fact, it does not model temporal structure at all—and, whereas it does require a significant amount of computation to estimate the speaker dependent bases, the separation process can easily be implemented in real time.

References

Cichocki, A., Zdunek, R., ichi Amari, S., 2006. Csiszar's divergences for nonnegative matrix factorization: Family of new algorithms. In: Lecture Notes in Computer Science. Vol. 3889. pp. 32–39.

Cooke, M. P., B. J. C. S. P., Shao, X., 2006. An audio-visual corpus for speech perception and automatic speech recognition. Journal of the Acoustical Society of America 120, 2421–2424.

Dhillon, I. S., Sra, S., 2005. Generalized nonnegative matrix approximations with bregman divergences. Tech. rep., University of Texas at Austin, Department of Computer Sciences.

Donoho, D., Stodden, V., 2003. When does non-negative matrix factorization give a correct decomposition into parts? In: Advances in Neural Information Processing Systems.

Eggert, J., Körner, E., 2004. Sparse coding and NMF. In: Neural Networks, IEEE International Conference on. Vol. 4. pp. 2529–2533.

Hoyer, P., 2002. Non-negative sparse coding. In: Neural Networks for Signal Processing, IEEE Workshop on. pp. 557–565.

Kim, D., Sra, S., Dhillon, I. S., 2007. Fast newton-type methods for the least squares nonnegative matrix approximation problem. In: Data Mining, Proceedings of SIAM Conference on.

Kompass, R., 2007. A generalized divergence measure for nonnegative matrix factorization. Neural Computation 19 (3), 780–791.

Kristjansson, T., Hershey, J., Olsen, P., Rennie, S., Gopinath, R., 2006. Superhuman multi-talker speech recognition: The IBM 2006 speech separation challenge system. In: International Conference on Spoken Language Processing (INTERSPEECH). pp. 97–100.

Lee, D. D., Seung, H. S., 1999. Learning the parts of objects by non-negative matrix factorization. Nature 401 (6755), 788–791.

Lee, D. D., Seung, H. S., 2000. Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems. pp. 556–562.

Lewicki, M. S., april 2002. Efficient coding of natural sound. Nature Neuroscience 5 (4), 356–363.

Lin, C.-J., 2007. Projected gradient methods for non-negative matrix factorization. Neural Computation (to appear).

Roweis, S. T., 2003. Factorial models and refiltering for speech separation and denoising. In: Eurospeech. pp. 1009–12.

Sajda, P., D. S. P. L., 2003. Recovery of constituent spectra using non-negative matrix factorization. In: Wavelets: Applications in Signal and Image Processing X, Proceedings of SPIE. Vol. 5207. pp. 321–331.

Schmidt, M. N., Olsson, R. K., 2006. Single-channel speech separation using sparse non-negative matrix factorization. In: International Conference on Spoken Language Processing (INTERSPEECH).

Srinivasan, S., Roman, N., Wang, D., 2006. Binary and ratio time-frequency masks for robust speech recognition. Speech Communication 48, 1486–1501.