Feature Space Reconstruction for Single-Channel Speech Separation

Mikkel N. Schmidt*, *Student Member, IEEE*,
Rasmus K. Olsson, *Student Member, IEEE*
Technical University of Denmark
Richard Petersens Plads, Bldg. 321
DK-2800 Kgs. Lyngby, Denmark
Email: mns@imm.dtu.dk
Fax: +45 45872599
Telephone: +45 45253888

Abstract

In this work we address the problem of separating multiple speakers from a single microphone recording. We formulate a linear regression model for estimating each speaker based on features derived from the mixture. The employed feature representation is a sparse, non-negative encoding of the speech mixture in terms of pre-learned speaker-dependent dictionaries. Previous work has shown that this feature representation by itself provides some degree of separation. We show that the performance is significantly improved when regression analysis is performed on the sparse, non-negative features.

I. Introduction

The cocktail-party problem can be defined as that of isolating or recognizing speech from an individual speaker in the presence of interfering speakers. An impressive feature of the human auditory system, this is essentially possible using only one ear, or, equivalently, listening to a mono recording of the mixture. It is an interesting and currently unsolved research problem to devise an algorithm which can mimic this ability.

A number of signal processing approaches have been based on learning speaker-dependent models on a training set of isolated recordings and subsequently applying a combination of these to the mixture. One possibility is to use a hidden Markov model (HMM) based on a Gaussian mixture model (GMM) for each speech source and combine these in a factorial HMM to separate a mixture [1]. Direct (naive) inference in such a model is not practical because of the dimensionality of the combined state space of the factorial HMM, necessitating some trick in order to speed up the computations. Roweis shows how to obtain tractable inference by exploiting the fact that in a log-magnitude time-frequency representation, the sum of speech signals is well approximated by the maximum. This is reasonable, since speech is sparsely distributed in the time-frequency domain. Recently, impressive results have been achieved by Kristjansson et al. [2] who devise an efficient method of inference that does not use the max-approximation. Based on a range of other approximations, they devise a complex system which in some situations exceeds human performance in terms of the error rate in a word recognition task.

Bach and Jordan [3] do not learn speaker dependent models but instead decompose a mixture by clustering the time-frequency elements according to a parameterized distance measure designed with the psychophysics of speech in mind. The algorithm is trained by learning the parameters of the distance measure from a training data set.

Another class of algorithms, here denoted ‘dictionary methods’, generally rely on learning a matrix factorization, in terms of a dictionary and its encoding for each speaker, from training data. The dictionary is a source dependent basis, and the method relies on the dictionaries of the sources in the mixture being sufficiently different. Separation of a mixture is obtained by computing the combined encoding using the concatenation of the source dictionaries. As opposed to the HMM/GMM based methods, this does not require a combinatorial search and leads to faster inference. Different matrix factorization methods can be conceived based on various a priori assumptions. For instance, independent component analysis and sparse decomposition, where the encoding is assumed to be sparsely distributed, have been proposed for single-channel speech separation [4], [5]. Another way to constrain the matrices is achieved through the assumption of non-negativity [6], [7], which is especially relevant when modeling speech in a magnitude spectrogram representation. Sparsity and non-negativity priors have been combined in sparse, non-negative matrix factorization [8] and applied to music and speech separation tasks [9], [10], [11].

In this work, we formulate a linear regression model for separating a mixture of speech signals based on features derived from a real-valued time-frequency representation of the speech. As a set of features, we use the encodings pertaining to dictionaries learned for each speaker using sparse, non-negative matrix factorization. The resulting maximum posterior estimator is linear in the observed mixture features and has a closed-form solution. We evaluate the performance of the method on synthetic speech mixtures by computing the signal-to-error ratio, which is the simplest, arguably sufficient, quality measure [12].

II. Methodology

The problem is to estimate P speech sources from a single microphone recording,

$$y(t) = \sum_{i=1}^{P} y_i(t), \tag{1}$$

where $y(t)$ and $y_i(t)$ are the time-domain mixture and source signals respectively. The separation is computed in an approximately invertible time-frequency representation, $\boldsymbol{Y} = \mathrm{TF}\left\{y(t)\right\}$, where $\boldsymbol{Y}$ is a real-valued matrix with spectral vectors as columns.

A. Linear estimator

In the following we describe a linear model for estimating the time-frequency representations of the sources in a mixture based on features derived from the mixture. The linear model reads,

$$\boldsymbol{Y}_i = \boldsymbol{W}_i^\top(\boldsymbol{X} - \boldsymbol{\mu}\boldsymbol{1}^\top) + \boldsymbol{m}_i\boldsymbol{1}^\top + \boldsymbol{N}, \tag{2}$$

where $\boldsymbol{Y}_i = \mathrm{TF}\left\{y_i(t)\right\}$ is the time-frequency representation of the i'th source, $\boldsymbol{W}_i$ is a matrix of weights, $\boldsymbol{X}$ is a feature matrix derived from $\boldsymbol{Y}$, $\boldsymbol{\mu}$ is the mean of the features, $\boldsymbol{m}_i$ is the mean of the i'th source and $\boldsymbol{N}$ is an additive noise term.

If we assume that the noise follows an i.i.d. normal distribution, $\mathrm{vec}(\boldsymbol{N}) \sim \mathcal{N}(\boldsymbol{0}, \sigma_n^2\boldsymbol{I})$, and put an i.i.d. zero mean normal prior over the weights, $\mathrm{vec}(\boldsymbol{W}_i) \sim \mathcal{N}(\boldsymbol{0}, \sigma_w^2\boldsymbol{I})$, the maximum posterior (MAP) estimator of the i'th source is given by

$$\hat{\boldsymbol{Y}}_i = \boldsymbol{\Gamma}_i\boldsymbol{\Sigma}^{-1}(\boldsymbol{X}^* - \boldsymbol{\mu}\boldsymbol{1}^\top) + \boldsymbol{m}_i\boldsymbol{1}^\top, \tag{3}$$

where $\boldsymbol{X}^*$ is the feature mapping of the test mixture $\boldsymbol{Y}^*$ and

$$\begin{aligned} \boldsymbol{\Gamma}_i &= \big(\boldsymbol{Y}_i - \boldsymbol{m}_i\boldsymbol{1}^\top\big)\big(\boldsymbol{X} - \boldsymbol{\mu}\boldsymbol{1}^\top\big)^\top, & (4)\\ \boldsymbol{\Sigma} &= \big(\boldsymbol{X} - \boldsymbol{\mu}\boldsymbol{1}^\top\big)\big(\boldsymbol{X} - \boldsymbol{\mu}\boldsymbol{1}^\top\big)^\top + \frac{\sigma_n^2}{\sigma_w^2}\boldsymbol{I}. & (5) \end{aligned}$$

Here, $\boldsymbol{X}$ is a matrix with feature vectors computed on a training mixture with mean $\boldsymbol{\mu}$, and $\boldsymbol{Y}_i$ is the corresponding time-frequency representation of the source with mean $\boldsymbol{m}_i$. For a detailed derivation of the MAP estimator, see e.g. Rasmussen and Williams [13].

When an isolated recording is available for each of the speakers, it is necessary to construct the feature matrix, $\boldsymbol{X}$, from synthetic mixtures. One way to exploit the available data would be to generate mixtures, $\boldsymbol{X}$, such that all possible combinations of time-indices are represented. However, the number of sources and/or the number of available time-frames would be prohibitively large.

A feasible approximation can be found in the limit of a large training set by making two additional assumptions: i) the features are additive, $\boldsymbol{X} = \sum_i^P \boldsymbol{X}_i$ with means $\boldsymbol{\mu}_i$, which is reasonable for, e.g., sparse features, and ii) the sources are independent such that all cross-products are negligible. Then,

$$\begin{aligned} \boldsymbol{\Gamma}_i &\approx \big(\boldsymbol{Y}_i - \boldsymbol{m}_i \boldsymbol{1}^\top\big)\big(\boldsymbol{X_i} - \boldsymbol{\mu}_i \boldsymbol{1}^\top\big)^\top, &(6)\\ \boldsymbol{\Sigma} &\approx \sum_{i=1}^{P} \big(\boldsymbol{X}_i - \boldsymbol{\mu}_i \boldsymbol{1}^\top\big)\big(\boldsymbol{X}_i - \boldsymbol{\mu}_i \boldsymbol{1}^\top\big)^\top. &(7)\end{aligned}$$

B. Features

In this work, two sets of feature mappings are explored. The first, and most simple, is to use the time-frequency representation itself as input to the linear model,

$$\boldsymbol{X}_i = \boldsymbol{Y}_i, \qquad \boldsymbol{X}^* = \boldsymbol{Y}^*. \tag{8}$$

A second, more involved, possibility is to use the encodings of a sparse, non-negative matrix factorization algorithm (SNMF) [8] as the features (see appendix A for a summary of SNMF). Possibly, other dictionary methods provide equally viable features.

In the SNMF method, the time-frequency representation of the i'th source is modelled as $\boldsymbol{Y}_i \approx \boldsymbol{D}_i \boldsymbol{H}_i$ where $\boldsymbol{D}_i$ is a dictionary matrix containing a set of spectral basis vectors, and $\boldsymbol{H}_i$ is an encoding which describes the amplitude of each basis vector at each time point. In order to use the method to compute features for a mixture, a dictionary matrix is first learned separately on a training set for each of the sources. Next, the mixture and the training data is mapped onto the concatenated dictionaries of the sources,

$$\boldsymbol{Y}_i \approx \boldsymbol{D}\boldsymbol{H}_i, \qquad \boldsymbol{Y}^* \approx \boldsymbol{D}\boldsymbol{H}^*, \tag{9}$$

where $\boldsymbol{D} = [\boldsymbol{D}_1, \ldots, \boldsymbol{D}_P]$. The encoding matrices, $\boldsymbol{H}_i$ and $\boldsymbol{H}^*$, are used as features,

$$\boldsymbol{X}_i = \boldsymbol{H}_i, \qquad \boldsymbol{X}^* = \boldsymbol{H}^*. \tag{10}$$

In previous work, the sources were estimated directly from these features as $\hat{\boldsymbol{Y}}_i = \boldsymbol{D}_i \boldsymbol{H}_i^*$ [11]. For comparison, we include this method in our evaluations. This method yields very good results when the sources, and thus the dictionaries, are sufficiently different from each other. In practice, however, this will not always be the case. In the factorization of the mixture, $\boldsymbol{D}_1$ will not only encode $\boldsymbol{Y}_1$ but also $\boldsymbol{Y}_2$ etc. This indicates that the encodings should rather be used as features in an estimator for each source.

III. Evaluation

The proposed speech separation method was evaluated on a subset of the GRID speech corpus [14] consisting of the first 4 male and first 4 female speakers (no. 1, 2, 3, 4, 5, 7, 11, and 15). The data was preprocessed by concatenating $T = 300$ s of speech from each speaker and resampling to $F_s = 8$ kHz. As a measure of performance, the signal-to-error ratio (SER) averaged across sources was computed in the time-domain. The testing was performed on synthetic 0 dB mixtures of two speakers, $T_{\text{test}} = 20$ s, constructed from all combinations of speakers in the test set.

In figures 1 and 2, the performance is shown for a collection of feature sets. The acronyms MAP-mel and MAP-SNMF refer to using the mel spectrum or the SNMF encoding as features, respectively. For reference, figures are provided for the basic SNMF approach as well [11]. The numeral suffix, '1' or '5', indicates whether using one or stacking five consecutive feature vectors, spaced 32 ms. The best performance is achieved for MAP-SNMF-5, reaching an $\simeq 1.2$ dB average improvement over the SNMF algorithm. It is noteworthy that the improvement is larger for the most difficult mixtures, those involving same-gender speakers.

In order to verify that the method is robust to changes in the relative gain of the signals in the mixtures, the performance was evaluated in a range of different target-to-interference ratios (TIR) (see figure 3). The results indicate that the method works very well even when the TIR is not known a priori. In figure 5, the performance is measured as a function of the available training data, indicating that the method is almost converged at 300 s.

The time-frequency representation was computed by normalizing the time-signals to unit power and computing the short-time Fourier transform (STFT) using 64 ms Hamming windows with 50% overlap. The absolute value of the STFT was then mapped onto a mel frequency scale using a publicly available toolbox [15] in order to reduce the dimensionality. Finally, the mel-frequency spectrogram was amplitude-compressed by exponentiating to the power p. By cross-validation we found that best results were obtained at $p = 0.55$ which gave significantly better results compared with, e.g., operating in the amplitude ($p = 1$) or the power ($p = 2$) domains (see figure 4). Curiously, this model prediction is similar to the empirically determined $p \approx 0.67$ exponent used in power law modelling of perceived loudness in humans, known as Stevens' Law, (see for example Hermansky [16]).

In the dictionary learning phase, the SNMF algorithm was allowed 250 iterations to converge from random initial conditions drawn from a uniform distribution on the unit interval. The number of dictionary atoms was fixed at $r = 200$ and the level of sparsity was chosen by cross-validation to $\lambda = 0.15$.

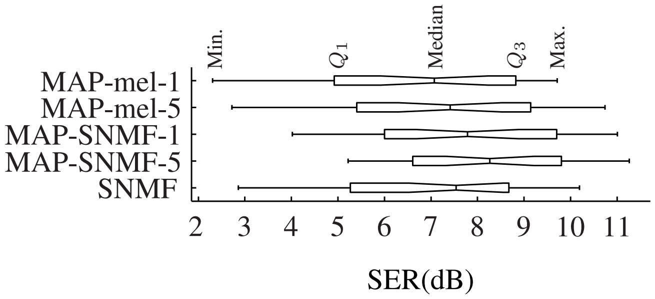

Fig. 1. The distribution of the signal-to-error (SER) performance of the method for all combinations of two speakers. The mel magnitude spectrogram (MAP-mel) and the SNMF encodings (MAP-SNMF) were used as features to the linear model. The results of using basic SNMF are given as a reference. The box plots indicate the extreme values along with the quartiles of the dB SER, averaged across sources.

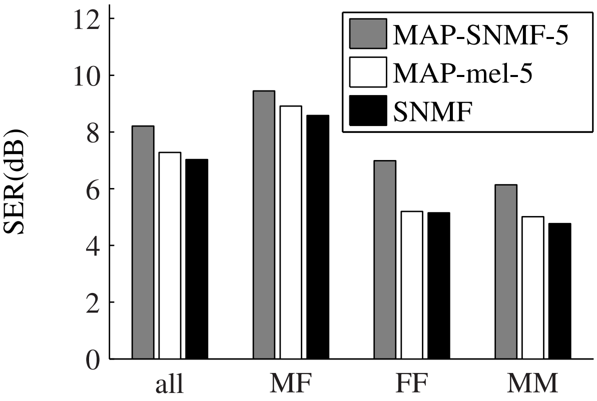

Fig. 2. The performance of the methods given as signal-to-error (SER) in dB, depending on the gender of the speakers. Male and female are identified by ‘M’ and ‘F’, respectively. The improvement of MAP-SNMF-5 over MAP-mel-5 and SNMF is largest in the most difficult (same-gender) mixtures.

When computing the encodings on the test mixtures, we found that non-negativity alone was sufficiently restrictive, hence $\lambda = 0$.

Time-domain reconstruction was performed by binary masking in the STFT spectrogram and subsequent inversion using the phase of the original mixture as described for example by Wang and Brown [17]. The phase errors incurred by this procedure are not severe due to the sparsity of speech in the spectrogram representation. Audio examples of the reconstructed speech are available online [18].

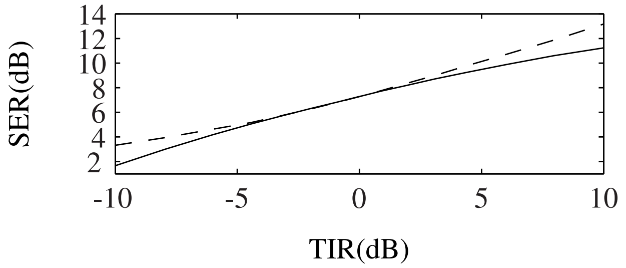

Fig. 3. The performance of the MAP-mel-5 algorithm given as the signal-to-error ratio (SER) of the target signal versus the target-to-interference ratio (TIR) of the mixture. The solid and dashed curves represent training on 0dB or the actual TIR of the test mixture, respectively. Clearly, the method is robust to a mismatch of the TIR between the training and test sets.

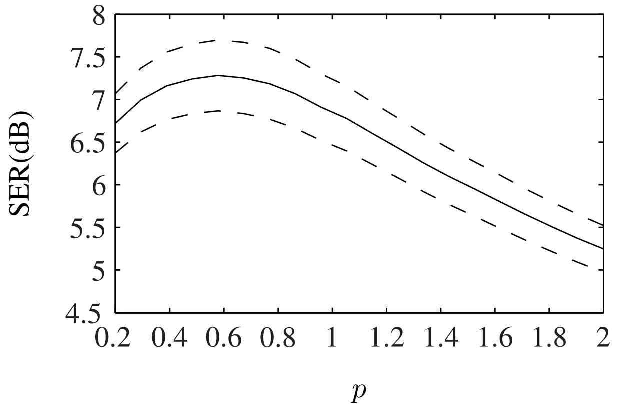

Fig. 4. The effect of amplitude compression on the performance of the MAP-mel-5 algorithm as measured in the signal-to-error ratio (SER). The optimal value of the exponent was found at $p \simeq 0.55$, in approximate accordance with Steven's power law for hearing. The dashed curve indicates the standard deviation of the mean.

IV. Discussion

The presented framework enjoys at least two significant advantages. First and foremost, computation in the linear model is fast. The estimation of the separation matrix is closed-form given the features, and the most time-consuming operation in the separation phase is a matrix product scaling with the dimensions of spectrogram and the number of features. Secondly, it is possible to fuse different features sets. Here, the spectrogram and sparse NMF were used, but many others could be imagined, possibly inspired by auditory models. The estimator integrates features across time, although the effect is relatively small, confirming previous reports that the inclusion of a dynamical model yields only marginal improvements [2], [19].

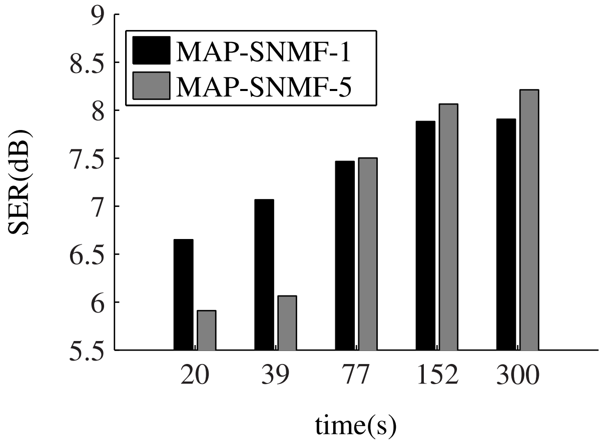

Fig. 5. The learning curve of the method, measured in signal-to-error ratio (SER), as a function of the size of the training set, depending on the complexity of the method.

ACKNOWLEDGMENT

During the research process, L. K. Hansen, J. Larsen and O. Winther administered advice and good suggestions.

APPENDIX

A. Sparse Non-negative Matrix Factorization

Let $\boldsymbol{Y} \geq \boldsymbol{0}$ be a non-negative data matrix. We model $\boldsymbol{Y}$ by

$$\boldsymbol{Y} = \boldsymbol{D}\boldsymbol{H} + \boldsymbol{N}, \tag{11}$$

where $\boldsymbol{N}$ is normal i.i.d. zero mean with variance σ_n^2. This gives rise to the likelihood function,

$$p(\boldsymbol{Y}|\boldsymbol{D},\boldsymbol{H}) \propto \exp\left(-\frac{|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{H}|_F^2}{2\sigma_n^2}\right), \tag{12}$$

where $|\cdot|_F$ denotes the Frobenius norm. We put a prior on $\boldsymbol{D}$ that is uniform over the part of the unit hyper-sphere lying in the positive orthant, i.e., $\boldsymbol{D}$ is non-negative and column-wise normalized. To obtain sparsity, the prior on $\boldsymbol{H}$ is assumed i.i.d. one-sided exponential, $p(\boldsymbol{H}) \propto \exp(-\beta|\boldsymbol{H}|_1)$, $\boldsymbol{H} \geq \boldsymbol{0}$, where $|\boldsymbol{H}|_1 = \sum_{ji} |h_{ji}|$. Now, the log-posterior can be written as

$$\log p(\boldsymbol{D},\boldsymbol{H}|\boldsymbol{Y}) \propto -\frac{1}{2}|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{H}|_F^2 - \lambda|\boldsymbol{H}|_1, \tag{13}$$
$$\text{s.t. } \boldsymbol{D} \geq \boldsymbol{0},\ |\boldsymbol{d}_j|_2 = 1,\ \boldsymbol{H} \geq \boldsymbol{0},$$

where $\boldsymbol{d}_j$ is the j'th column vector of $\boldsymbol{D}$.

The log-posterior can be seen as a quadratic cost function augmented by an L_1 norm penalty term on the coefficients in $\boldsymbol{H}$. The hyper-parameter $\lambda = \beta\sigma_n^2$ controls the degree of sparsity. A maximum posterior (MAP) estimate can be computed by optimizing (13) with respect to $\boldsymbol{D}$ and $\boldsymbol{H}$.

Eggert and Körner [8] derive a simple algorithm for computing this MAP estimate based on alternating multiplicative updates of $\boldsymbol{D}$ and $\boldsymbol{H}$

$$\boldsymbol{H} \quad \leftarrow \quad \boldsymbol{H} \bullet \frac{\bar{\boldsymbol{D}}^\top \boldsymbol{Y}}{\bar{\boldsymbol{D}}^\top \widetilde{\boldsymbol{Y}} + \boldsymbol{\Lambda}}, \tag{14}$$

$$\boldsymbol{d}_j \quad \leftarrow \quad \bar{\boldsymbol{d}}_j \bullet \frac{\sum_i h_{ji} \left[\boldsymbol{y}_i + (\widetilde{\boldsymbol{y}}_i^\top \bar{\boldsymbol{d}}_j)\bar{\boldsymbol{d}}_j\right]}{\sum_i h_{ji} \left[\widetilde{\boldsymbol{y}}_i + (\boldsymbol{y}_i^\top \bar{\boldsymbol{d}}_j)\bar{\boldsymbol{d}}_j\right]}, \tag{15}$$

where $\widetilde{\boldsymbol{Y}} = \bar{\boldsymbol{D}}\boldsymbol{H}$, $\bar{\boldsymbol{D}}$ is the column-wise normalized dictionary matrix, $\boldsymbol{\Lambda}$ is a matrix with elements λ, and the bold operators indicate pointwise multiplication and division.

REFERENCES

[1] S. T. Roweis, "One microphone source separation," in *Advances in Neural Information Processing Systems*, 2000, pp. 793–799.

[2] T. Kristjansson, J. Hershey, P. Olsen, S. Rennie, and R. Gopinath, "Super-human multi-talker speech recognition: The IBM 2006 speech separation challenge system," in *International Conference on Spoken Language Processing (INTERSPEECH)*, 2006, pp. 97–100.

[3] F. R. Bach and M. I. Jordan, "Blind one-microphone speech separartion: A spectral learning approach," in *Advances in Neural Information Processing Systems*, 2005, pp. 65–72.

[4] G. J. Jang and T. W. Lee, "A maximum likelihood approach to single channel source separation," *Journal of Machine Learning Research*, vol. 4, pp. 1365–1392, 2003.

[5] B. A. Pearlmutter and R. K. Olsson, "Algorithmic differentiation of linear programs for single-channel source separation," in *Machine Learning and Signal Processing, IEEE International Workshop on*, 2006.

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

[7] P. Smaragdis, "Discovering auditory objects through non-negativity constraints," in *Statistical and Perceptual Audio Processing (SAPA)*, 2004.

[8] J. Eggert and E. Körner, "Sparse coding and NMF," in *Neural Networks, IEEE International Conference on*, vol. 4, 2004, pp. 2529–2533.

[9] T. Virtanen, "Sound source separation using sparse coding with temporal continuity objective," in *International Computer Music Conference, ICMC*, 2003.

[10] L. Benaroya, L. M. Donagh, F. Bimbot, and R. Gribonval, "Non negative sparse representation for wiener based source separation with a single sensor," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 2003, pp. 613–616.

[11] M. N. Schmidt and R. K. Olsson, “Single-channel speech separation using sparse non-negative matrix factorization,” in *International Conference on Spoken Language Processing (INTERSPEECH)*, 2006.

[12] D. Ellis, “Evaluating speech separation systems,” in *Speech Separation by Humans and Machines*, P. Divenyi, Ed. Kluwer Academic Publishers, ch. 20, pp. 295–304.

[13] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[14] M. P. Cooke, J. Barker, S. P. Cunningham, and X. Shao, “An audio-visual corpus for speech perception and automatic speech recognition,” *submitted to JASA*.

[15] D. P. W. Ellis. (2005) PLP and RASTA (and MFCC, and inversion) in Matlab. [Online]. Available: http://www.ee.columbia.edu/∼dpwe/resources/matlab/rastamat

[16] H. Hermansky, “Perceptual linear predictive (plp) analysis of speech,” *The Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[17] D. L. Wang and G. J. Brown, “Separation of speech from interfering sounds based on oscillatory correlation,” *IEEE-NN*, vol. 10, no. 3, p. 684, 1999.

[18] M. N. Schmidt and R. K. Olsson. (2007) Audio samples relevant to this letter. [Online]. Available: http://mikkelschmidt.dk/spletters2007

[19] P. Smaragdis, “Convolutive speech bases and their application to supervised speech separation,” *IEEE Transaction on Audio, Speech and Language Processing - to appear*, 2007.