EFFICIENT MODEL-BASED SPEECH SEPARATION AND DENOISING USING NON-NEGATIVE SUBSPACE ANALYSIS

Steven J. Rennie, John R. Hershey, and Peder A. Olsen

IBM Thomas J. Watson Research Center

{sjrennie,jrhershe,pederao}@us.ibm.com

ABSTRACT

We present a new probabilistic architecture for analyzing composite non-negative data, called Non-negative Subspace Analysis (NSA). The NSA model provides a framework for understanding the relationships between sparse subspace and mixture model based approaches, and encompasses a range of models, including Sparse Non-negative Matrix Factorization (SNMF) [1] and mixture-model based analysis as special cases. We present a convenient instantiation of the NSA model, and an efficient variational approximate learning and inference algorithm that combines the advantages of SNMF and mixture model-based approaches. Preliminary recognition results on the Pascal Speech Separation Challenge 2006 test set [2], based on NSA separation results, are presented. The results fall short of those achieved by Algonquin [3], a state-of-the-art mixture-model based method, but considering that NSA runs an order of magnitude faster, the results are impressive. NSA outperforms SNMF in terms of word error rate (WER) on the task by a significant margin of over 9% absolute.

Index Terms— Non-negative Subspace Analysis (NSA), Speech Separation, Variational Expectation-Maximization (GEM), Robust Speech Recognition, Sparse Non-negative Matrix Factorization (SNMF)

1. INTRODUCTION

Model-based speech separation and denoising has been a heavily researched topic in robust speech recognition in recent years. A common approach is to model each source using a mixture model. In this approach, exact inference scales exponentially with the number of sources, because all possible mixture combinations must be explored.

Iterative approximate inference schemes, such as variational methods [3], have been applied to make inference linear rather than exponential in the number sources for mixture-based models, and produced some very impressive results. Such approaches are in practice still computationally expensive, however, because the required computation per iteration and number of required iterations is generally quite significant. Approximate source and interaction models, including band quantized models [4] and the "max-model" in the log spectrum [5], can be used to greatly reduce the amount of computation per state combination, but exact inference still scales exponentially with the number of sources.

Subspace-based approaches such as non-negative matrix factorization [6, 1, 7], on the other hand, are extremely computationally efficient. Source subspaces can be learned on separated data and concatenated to analyze composite data without explicitly considering the possible "state combinations" of the sources. Subspace and sparse analysis representations are a hot topic in signal processing right now, but despite this, relatively little work exists that directly compares the speed and performance of sparse subspace and mixture model based methods or explores their relationship.

In this paper, we present a new probabilistic architecture for analyzing composite non-negative data, called Non-negative Subspace Analysis (NSA). NSA provides a framework for understanding the relationships between sparse subspace and mixture model based approaches, and encompasses a range of models, including Sparse Non-negative Matrix Factorization (SNMF) [1] and mixture-model based analysis as special cases. We present a convenient instantiation of the NSA model, and an efficient variational approximate learning and inference algorithm that combines some of the advantages of NMF and mixture model-based approaches.

Preliminary speech recognition results on the Pascal Speech Separation Challenge 2006 test set [2], based on NSA separation results, are presented. The results fall short of those achieved by Algonquin [3], a state-of-the-art mixture-model based method, but considering that NSA runs an order of magnitude faster, the results are impressive. NSA outperforms SNMF in terms of word error rate (WER) on the task by a significant margin of over 9% absolute.

2. NON-NEGATIVE SUBSPACE ANALYSIS

We model the probability density of non-negative composite vector data $\boldsymbol{y}$ as a superposition of non-negative probabilistic subspaces:

$$p(\boldsymbol{y}) = \int_{\boldsymbol{v}} \int_{\boldsymbol{c}} \sum_{\boldsymbol{a}} p(\boldsymbol{y}|\boldsymbol{v},\boldsymbol{c}) \cdot \prod_s p(\boldsymbol{a}_s) \prod_n p(c_{sn}|\boldsymbol{a}_s) p(\boldsymbol{v}_{sn}|\boldsymbol{a}_s), \quad (1)$$

where c_{sn} and $\boldsymbol{v}_{sn}$ are random variables representing the coefficient and basis vector of component n of subspace s, respectively, and $\boldsymbol{a}_s$ encodes the collective binary activity/inactivity of the components of subspace s. If the component activations are constrained such that exactly one component is active in each subspace, the representation reduces to a mixture model based data decomposition.

In this paper we will assume that $\boldsymbol{y}$ is composed from a linear combination of subspace vectors, plus zero mean diagonal covariance gaussian noise

$$p(\boldsymbol{y}|\boldsymbol{v},\boldsymbol{c}) = \mathcal{N}(\boldsymbol{y}; \sum_{sn} c_{sn}\boldsymbol{v}_{sn}, \boldsymbol{\Psi}), \quad (2)$$

that the component activations of each subspace are independent

$$p(\boldsymbol{a}_s) = \prod_n \pi_{sn}^{a_{sn}} (1-\pi_{sn})^{1-a_{sn}}, \quad (3)$$

and model the conditional distribution of each basis vector component given that it is active as a diagonal-covariance gaussian

$$p(\boldsymbol{v}_{sn}|a_{sn}=1)=\mathcal{N}(\boldsymbol{v}_{sn};\boldsymbol{\mu}_{sn},\boldsymbol{\Sigma}_{sn}), \tag{4}$$

where $\|\boldsymbol{\mu}_{sn}\|_2=1$. We model the distribution of the coefficients as

$$p(c_{sn}|a_{sn})=\begin{cases}\mathcal{N}(c_{sn};\alpha_s+\beta_{sn},\tau_{sn}), & a_{sn}=1\\ \lambda_{sn}\exp(-\lambda_{sn}c_{sn}), & a_{sn}=0\end{cases} \tag{5}$$

where $\lambda_{sn}\gg 0$. Note that inactive components can have non-zero coefficients, but since $\lambda_{sn}\gg 0$, they contribute negligibly to the generation of the observed data. As such, the conditional distribution of inactive basis vectors can be somewhat arbitrarily set. A setting that will prove very convenient for efficient learning and inference is $p(\boldsymbol{v}_{sn}|a_{sn}=0)=p(\boldsymbol{v}_{sn}|a_{sn}=1)$. Note that the conditional mean of active coefficients consists of a subspace specific "gain" parameter α_s, and a subspace and component-specific gain, β_{sn}.

This instantiation of the NSA model is related to Sparse Non-negative Matrix Factorization (SNMF) with a quadratic primary objective [1]. In SNMF, the objective $\|Y-VC\|_{\mathcal{F}}^2+\lambda\sum_t\|c[t]\|_1=\sum_t(\boldsymbol{y}[t]-\sum_n c_n[t]\boldsymbol{v}_n)^2+\lambda\sum_n|c_n[t]|$ s.t. to $\{c_n[t]\},\{v_{nd}\}>0$, is optimized to find a sparse representation of each column $\boldsymbol{y}[t]$ of Y in terms of the basis set $\{\boldsymbol{v}_n\}$. This objective is equal to the negative log probability of the columns of Y under the assumption that the basis coefficient priors are exponentially distributed with mean $\frac{1}{\lambda}$, and unit variance gaussian noise in the representation. The presented NSA model differs and extends upon SNMF in several ways. In NSA, information about the relative scale of each basis component is represented independently of its activation characteristics, which makes it straightforward to utilize any known information about the component activations or gains, and to extend the model. The activation priors, for example, can be made context-dependent to better model the characteristics of highly structured source signals such as speech.

Another important property of the NSA model is that the component vectors are random variables rather than parameters. The data is composed not from basis vectors, but from basis *distributions* to better represent the underlying probability density of the hidden source represented by each subspace. This is particularly important when the basis representation is sparse, because otherwise the probability distribution of each source would be confined to a hyperplane of dimension much lower than the data vector. It also facilitates the computation of basis vector *posteriors*, that can be used to recover context-dependent estimates of the hidden sources they represent.

It bears noting that the Probabilistic Sparse Non-negative matrix Factorization (PSNMF) model presented in [8] also differs from NSA in many important respects. In PSNMF, the component priors are modelled as zero-mean gaussians with unit variance, the coefficient priors as uniformly distributed, and the number of active components as multinomial distributed. This model is designed specifically for blind analysis, whereas NSA has been designed to learn and utilize source specific characteristics to separate composite signals, whose pieces can optionally be trained on isolated data. The multinomial-distributed activation prior in PSNMF is very general but is not amenable to continuous relaxation, and so even approximate inference techniques are computationally intensive.

In contrast, in the NSA model presented here the component activations are assumed to be independent, which make the model amenable to continuous relaxation. Note that if the activation priors were constrained to be equal in the NSA model, then the prior on the number of active coefficients would be binomial-distributed. The mean number of active coefficients in a subspace N_{a_s} can be upper-bounded to enforce sparsity by upper-bounding the probability of activation by $\overline{\pi}_{a_s}$ so $E[N_{a_s}]\leq N_s\overline{\pi}_{a_s}$, where N_s is the number of components in subspace s. For $\overline{\pi}_{a_s}<\frac{1}{2}$, which is always the case for sparse representations, $Var[N_{a_s}]\leq N_s\overline{\pi}_{a_s}(1-\overline{\pi}_{a_s})\leq N_s\overline{\pi}_{a_s}$. Therefore despite the independence assumption on the activity of the components in the presented NSA model, the framework provides a means of controlling the sparseness of the representation.

3. LEARNING AND INFERENCE

Exact inference is generally intractable in the presented NSA model. The component activations are discrete binary random variables and so inference scales exponentially $O(2^C)$ in total number of components $C=\sum_s N_s$. One option is to apply iterative approximate inference techniques such as variational methods or the sum-product algorithm [3, 9] to estimate the component activations. Such approaches can be designed to scale linearly in the number of components, but will require that the activations be updated iteratively, which ignores important correlations in the component activations during the optimization, and is quite computationally expensive in practice. Here we achieve tractable learning and inference via an approximate expectation-maximization (EM) algorithm that solves a continuous relaxation of this expensive inference problem during the E-Step with an efficient variational algorithm.

3.1. E-Step

We avoid the expensive task of inferring the component activations by marginalizing them out, and then approximating the marginal prior of the coefficients as exponentially distributed:

$$\begin{aligned}p(c_{sn}) &= \pi_{sn}\mathcal{N}(c_{sn};\alpha_s+\beta_{sn},\tau_{sn})\\ &\quad+(1-\pi_{sn})\lambda_{sn}\exp(-\lambda_{sn}c_{sn}),\\ &\approx \chi_{sn}\exp(-\chi_{sn}c_{sn})\equiv\tilde{p}(c_{sn}),\end{aligned} \tag{6}$$

where χ_{sn} is obtained by moment-matching

$$\begin{aligned}\chi_{sn} &= (E[c_{sn}])^{-1}\\ &= (\pi_{sn}(\alpha_s+\beta_{sn})+(1-\pi_{sn})/\lambda_{sn})^{-1}.\end{aligned} \tag{7}$$

The approximation is reasonable because $\pi_{sn}\ll 1-\pi_{sn}$ when the representation is designed to be sparse. Given this approximation, the joint distribution of the component coefficients, vectors, and the observed data vector is:

$$\tilde{p}(\boldsymbol{v},\boldsymbol{c},\boldsymbol{y})=p(\boldsymbol{y}|\boldsymbol{c},\boldsymbol{v})p(\boldsymbol{v})\tilde{p}(\boldsymbol{c}). \tag{8}$$

Given the observation, $\boldsymbol{y}$, the hidden variables $\boldsymbol{v}$ and $\boldsymbol{c}$ are non-linearly related, and the posterior distribution, $\tilde{p}(\boldsymbol{v},\boldsymbol{c}|\boldsymbol{y})$, is non-gaussian and computationally intractable to estimate. However, the model is convex in $\boldsymbol{c}$ given $\boldsymbol{v}$ and vice versa. We therefore approximate the posterior distribution of $\boldsymbol{c}$ and $\boldsymbol{v}$ with a variational surrogate distribution with the following factorized form:

$$q(\boldsymbol{c},\boldsymbol{v})=q(\boldsymbol{c})q(\boldsymbol{v})=q(\boldsymbol{c})\prod_d p(\boldsymbol{v}_d) \tag{9}$$

where $\boldsymbol{v}_d=\mathrm{vec}(\{v_{snd}\})$ is the vector formed from the elements of the component vectors $\{\boldsymbol{v}_{sn}\}$ in dimension d. The factorization over the dimensions of the basis vectors follows from the diagonal covariance of the basis and observation priors. Note that in $\tilde{p}(\boldsymbol{v},\boldsymbol{c}|\boldsymbol{y})$, the basis vectors $\boldsymbol{v}_d$ given $\boldsymbol{c}$ for each dimension d are correlated, as are

the basis coefficients $\boldsymbol{c}$ given the basis vectors, $\boldsymbol{v}$. Therefore we take the variational posteriors of $\boldsymbol{c}$ and $\boldsymbol{v}_d$ to be full-covariance gaussians:

$$q(\boldsymbol{c}) = \mathcal{N}(\boldsymbol{c}; \boldsymbol{\eta_c}, \boldsymbol{\Omega_c}) \tag{10}$$

$$q(\boldsymbol{v}_d) = \mathcal{N}(\boldsymbol{v}_d; \boldsymbol{\zeta}_{\boldsymbol{v}_d}, \boldsymbol{\Gamma}_{\boldsymbol{v}_d}) \tag{11}$$

The proposed form of the variational surrogate preserves the predominant structural properties of the true posterior, and leads to an approximate E-Step that iteratively optimizes the highly correlated subspaces of the hidden variables.

To identify q, we minimize the KL divergence between the surrogate posterior and the joint distribution of the random variables of the model. This correspondingly minimizes the KL divergence between the surrogate and true posterior distribution of the hidden variables of the model, and allows us to lower bound the probability of each data vector, and the collection of data vectors:

$$\begin{aligned}
\sum_t \log p(\boldsymbol{y}[t]) &= \sum_t \log \sum_{\boldsymbol{v}[t],\boldsymbol{c}[t]} \tilde{p}(\boldsymbol{v}[t], \boldsymbol{c}[t], \boldsymbol{y}[t]), \\
&\geq \sum_t \sum_{\boldsymbol{v}[t],\boldsymbol{c}[t]} q(\boldsymbol{v}[t], \boldsymbol{c}[t]) \log \frac{\tilde{p}(\boldsymbol{v}[t], \boldsymbol{c}[t], \boldsymbol{y}[t])}{q(\boldsymbol{v}[t], \boldsymbol{c}[t])}, \\
&= -\sum_t D(q(\boldsymbol{v}[t], \boldsymbol{c}[t]) \,||\, \tilde{p}(\boldsymbol{v}[t], \boldsymbol{c}[t] | \boldsymbol{y}[t])) + \sum_t \log p(\boldsymbol{y}[t]).
\end{aligned} \tag{12}$$

Exploiting the conditional independencies of the NSA model, and the factorized form of the q, we arrive at the following set of updates that may be iterated to identify the parameters of q:

$$\boldsymbol{\Gamma}_{\boldsymbol{v}_d}^{-1} = \boldsymbol{\Sigma}_d^{-1} + \psi_d^{-1}(\boldsymbol{\eta_c}\boldsymbol{\eta_c}^T + \boldsymbol{\Omega_c}) \tag{13}$$

$$\frac{\partial D_{q||p}}{\partial \boldsymbol{\zeta}_{\boldsymbol{v}_d}} = -\boldsymbol{\Gamma}_{\boldsymbol{v}_d}^{-1}\boldsymbol{\zeta}_{\boldsymbol{v}_d} + \boldsymbol{\Sigma}_d^{-1}\boldsymbol{\mu}_d + \boldsymbol{\eta_c}\psi_d^{-1}y_d \tag{14}$$

$$\boldsymbol{\zeta}^i_{\boldsymbol{v}_d,sn} = \boldsymbol{\zeta}^{i-1}_{\boldsymbol{v}_d,sn} \cdot \frac{(\partial D_{q||p}/\partial \boldsymbol{\zeta}_{\boldsymbol{v}_d})_{sn+}}{(\partial D_{q||p}/\partial \boldsymbol{\zeta}_{\boldsymbol{v}_d})_{sn-}} \tag{15}$$

$$\boldsymbol{\Omega_c}^{-1} = \sum_d \psi_d^{-1}(\boldsymbol{\zeta}_{\boldsymbol{v}_d}\boldsymbol{\zeta}_{\boldsymbol{v}_d}^T + \boldsymbol{\Gamma}_{\boldsymbol{v}_d}) \tag{16}$$

$$\frac{\partial D_{q||p}}{\partial \boldsymbol{\eta_c}} = -\boldsymbol{\Omega_c}^{-1}\boldsymbol{\eta_c} + \sum_d \boldsymbol{\zeta}_{\boldsymbol{v}_d}\psi_d^{-1}y_d - \boldsymbol{\lambda} \tag{17}$$

$$\boldsymbol{\eta}^i_{\boldsymbol{c},sn} = \boldsymbol{\eta}^{i-1}_{\boldsymbol{c},sn} \cdot \frac{(\partial D_{q||p}/\partial \boldsymbol{\eta_c})_{sn+}}{(\partial D_{q||p}/\partial \boldsymbol{\eta_c})_{sn-}} \tag{18}$$

where $D_{q||p} = D(q(\boldsymbol{v}[t], \boldsymbol{c}[t]) \,||\, \tilde{p}(\boldsymbol{v}[t], \boldsymbol{c}[t], \boldsymbol{y}[t]))$, $\boldsymbol{\lambda} = \text{vec}(\{\lambda_{sn}\})$, and the notation $()_{sn+}$ and $()_{sn-}$ denotes the positive and negative terms of the snth component of the vector argument, respectively. Multiplicative updates for the elements of $\boldsymbol{\zeta}_{\boldsymbol{v}_d}$ and $\boldsymbol{\eta_c}$ are used to enforce non-negativity, which is a common approach to optimizing NMF algorithms [6]. These updates are recursed during each iteration of the variational updates until convergence. The convergence of such updates has not been proved, but in practice this has not been an issue.

The algorithm scales quadratically with the total number of components $C = \sum_s N_s$ and linearly in the number of dimensions D as $O(DC^2)$, but in practice, the initial $\boldsymbol{\eta_c}$ update is $O(DC)$ because $\boldsymbol{\Gamma}_{\boldsymbol{v}_d}$ is initialized to be diagonal, and the number of components being considered can be pruned down to $C' \ll C$ after this initial update, with negligible loss in performance. In our experiments with $C = 512$, for example, $C' < 50$ for all test cases when components contributing less than 0.01% of the reconstruction intensity were pruned away after the first $\boldsymbol{\eta}_{c_t}$ update. This sped up the algorithm substantially. The performance impact of more aggressive pruning has not yet been investigated.

Note that $\boldsymbol{\Omega_c}^{-1}$, the precision of the current coefficient estimates $\boldsymbol{\eta_c}$, reshapes the optimization surface of the component vectors $\boldsymbol{\zeta}_{\boldsymbol{v}_d}$, and similarly, the component vector precisions $\{\boldsymbol{\Gamma}_{\boldsymbol{v}_d}^{-1}\}$ affect the gradient direction of $\boldsymbol{\eta_c}$.

3.2. M-Step

In the M-Step, the variational lower bound on the probability of the observed data (12) is maximized w.r.t. the parameters of the NSA model. The component vector parameter updates are given by:

$$\boldsymbol{\mu}_{sn} = \sum_t \pi'_{sn}[t]\boldsymbol{\zeta}_{\boldsymbol{v}_{sn}[t]} \tag{19}$$

$$\sigma^2_{sn,dd} = \sum_t \pi'_{sn}[t]((\mu_{sn,d} - \zeta_{\boldsymbol{v}_{sn},d[t]})^2 + \gamma_{\boldsymbol{v}_{sn}[t],dd}) \tag{20}$$

where $\pi'_{sn}[t] = p(a_{sn}[t] = 1|\boldsymbol{y}[t])$ is the posterior probability that the component n of subspace s is active. Here we estimate the component activation posteriors by simply computing the probability that each component is active/inactive given the posterior estimate of that component's coefficient:

$$\begin{aligned}
p(a_{sn}[t]|\boldsymbol{y}[t]) &\approx p(a_{sn}[t]|c_{sn}[t] = \eta_{c_{sn}[t]}) \\
&\propto p(a_{sn}[t])p(c_{sn}[t] = \eta_{c_{sn}[t]}|a_{sn}[t])
\end{aligned} \tag{21}$$

The coefficient parameter updates are given by:

$$\lambda_{n,s} = \left(\sum_t (1 - \pi'_{sn}[t])\eta_{c_{sn}[t]}\right)^{-1} \tag{22}$$

$$\alpha_s = \sum_{n,t} \pi'_{sn}[t](\eta_{c_{sn}[t]} - \beta_{sn}) \tag{23}$$

$$\beta_{sn} = \sum_t \pi'_{sn}[t](\eta_{c_{sn}[t]} - \alpha_s) \tag{24}$$

$$\tau_{sn} = \sum_t \pi'_{sn}[t]((\eta_{c_{sn}[t]} - (\beta_{sn} + \alpha_s))^2 + \omega_{c_{sn}[t]}) \tag{25}$$

Note that the $\alpha_s + \beta_{sn}$ representation of the active gain mean of the component coefficients of each subspace is under-determined. During learning α_s is fixed and β_{sn} is learned. At test time α_s can be adapted to re-normalize each source subspace to the test data.

4. EXPERIMENTS

The "same gender" and "different gender" subsets of the Pascal 2006 Speech Separation Challenge (SSC) test set [2], comprised of test utterances containing two talkers speaking simultaneously—synthetic mixtures generated from the Grid Corpus [10]—were used as a basis for evaluating the proposed NSA algorithm.

So that we could directly compare the performance and execution speed of NSA to Algonquin [11], a state-of-the-art source separation

Method	Algonquin	NSA	NSA$^-$	SNMF [7]
SG	25.7	41.6	50.7	53.0
DG	21.5	30.6	40.3	37.8
Overall	23.6	36.1	45.5	45.4

Table 1. Word error rate (WER) performance as a function of front-end separation algorithm, on the same gender (SG) and different gender (DG) subsets of the SSC test set. Algonquin, a mixture model-based separation method [11], outperforms the other approaches, which are subspace-based, but takes an order of magnitude more computation time. Non-negative subspace analysis (NSA), the algorithm proposed here, outperforms sparse non-negative matrix factorization (SNMF) on the task by more than 9% absolute.

method that models each speaker using a mixture model, NSA models for the sources were not learned but instead derived from learned mixture models. Speaker-dependent, 256 component diagonal covariance gaussian mixture models (GMMs) of speech, trained on 319 dimensional high-resolution log power spectrum features, derived from hamming-windowed 40 ms segments overlapped by 15 ms taken from the SSC training set, were used in all of our experiments. Whereas Algonquin operates on log spectral (or cepstral) features, NSA was applied in the power spectral domain, where the interaction between the sources is approximately linear and the features are non-negative, as assumed by the NSA model.

Speaker subspaces were generated from their respective log domain GMMs by moment matching to generate corresponding GMMs in the power spectral domain, and then normalizing the emission distributions to generate basis vector priors. The activation priors were directly taken as the mixture component priors. The identification and gain of the speakers was first estimated using the system described in [11]. The SSC test utterances were then denoised using Algonquin or NSA, and finally passed the recognition system described in [11], which does speaker-dependent labelling.

Table 1 summarizes the word error rate (WER) recognition results obtained on the same gender and different gender subsets of the SSC task by Algonquin and NSA. The results obtained using NSA but with the component vectors held fixed to their priors, denoted by NSA$^-$, are also depicted, as are the results obtained in [7] when SNMF was applied to the task. Looking at the results, we can see that Algonquin outperforms NSA on these tasks overall by over 10% absolute, but the NSA result is nevertheless impressive considering that it takes an order of magnitude less computation time than Algonquin. The results obtained by NSA are in turn more than 9% better overall than those obtained by NSA$^-$ and SNMF. NSA models the component vectors as random variables rather than parameters, and propagates uncertainty back and forth when iterating between estimating the coefficient and vector posteriors. This improves the quality of the reconstructed speech estimates and the recognition result.

5. FUTURE WORK

The initial results obtained using the NSA algorithm presented here are promising. Several important directions of future investigation remain. The experiments described here adapted mixture models into NSA subspace models, and from the log to power spectral domain by simple moment matching. Better results can surely be obtained by utilizing learned NSA models.

An additional and promising direction of future investigation is to make the component activation priors context-dependent. More specifically, we are excited about the prospect of using NSA to do noise and secondary speech robust feature labelling in our existing speech recognition systems, and developing fast multi-talker speech recognition systems based on NSA.

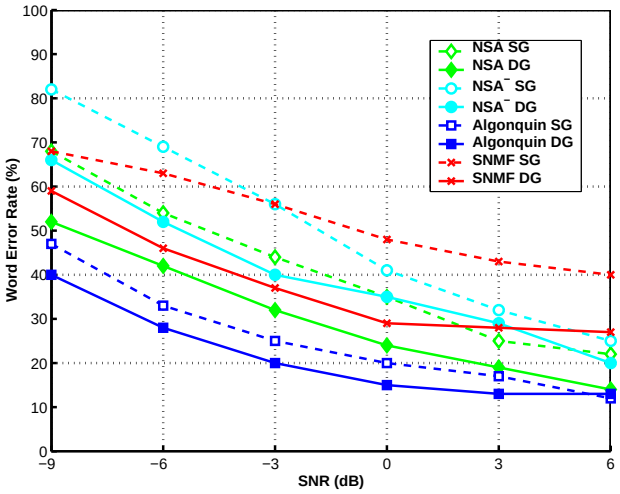

Fig. 1. Word error rate (WER) performance as a function of front-end separation algorithm and SNR on the same gender (SG) and different gender (DG) subsets of the SSC test set. NSA consistently outperforms SNMF over the task.

6. REFERENCES

[1] Patrik O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 14571469, 2004.

[2] Martin Cooke and Tee-Won Lee, "Interspeech speech separation challenge," `http://www.dcs.shef.ac.uk/~martin/SpeechSeparationChallenge.htm`, 2006.

[3] B.J. Frey, T. Kristjansson, L. Deng, and A. Acero, "Learning dynamic noise models from noisy speech for robust speech recognition," in *NIPS*, 2001.

[4] E. Bocchieri, "Vector quantization for the efficient computation of continuous density likelihoods. proceedings of the international conference on acoustics," in *ICASSP*. IEEE, 1993, vol. II, pp. 692–695.

[5] S. Roweis, "Factorial models and refiltering for speech separation and denoising," *Eurospeech*, pp. 1009–1012, 2003.

[6] D.Lee and H.Seung, "Algorithms for non-negative matrix factorization," in *NIPS*, 2000.

[7] Mikkel N. Schmidt and Rasmus K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *Interspeech*, sep 2006.

[8] D. Dueck and B.J. Frey, "Probabilistic sparse matrix factorization," in *PSI TR 2004.023*, 2004.

[9] B. J. Frey F. R. Kschischang and H-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47:2, 2001.

[10] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *Journal of the Acoustical Society of America*, vol. 120, pp. 2421–2424, 2006.

[11] T. Kristjansson, J. R. Hershey, P. A. Olsen, S. Rennie, and R. Gopinath, "Super-human multi-talker speech recognition: The IBM 2006 speech separation challenge system," in *ICSLP*, 2006.