Integrating LifeLogs by ICA structural learning

Andreas Brinch Nielsen
Informatics and Mathematical Modelling
Technical University of Denmark
abn@imm.dtu.dk

Lars Kai Hansen
Informatics and Mathematical Modelling
Technical University of Denmark
lkh@imm.dtu.dk

Abstract

The classic LifeLog is a data structure for accumulation of the total personal information flow. While earlier research on LifeLogs have focused on the individual level we are interested in *collective* properties of multipled LifeLogs. In particular we focus on the integration of LifeLog audio components for a small group of coworkers occupying a simple indoor environment. We consider a scenario in which each participant is wired with a single microphone. We approach the organization of the set of recordings as a cocktail party problem with variable mixing matrix. To identify short term social interaction patterns we estimate sparse ICA separation matrices and use the structure to make inferences: A 'non-zero' element indicates that a speaker is present in a given microphone.

1 Introduction

'Total recall' systems like LifeLogs[3], MylifeBits[8], and LifeStreams[7] have conventionally been framed in a personal computing context: How can I organize and retrieve the total information flow around me? Audio is an important aspect of these systems. Ellis and Lee proposed an audio segmentation system that can assist retrieval and summarization of personal audio archives [6, 5], while Kern et al., described a system for localization and context detection based on ambient sound [12].

From an organizational point of view, however, there is much to be gained from *integrating personal archives*. A system for measuring social interactions among elderly in a nursing home can potentially improve quality of service and detect complex interactions and potential conflicts as discussed by Chen et al., [2]. Eagle and Pentland have recently reported on a large experiment involving data collection from 100 mobile phones. The collection protocol consists of meta data related to the mobile, not the actual sound. However, the meta data is still rich enough to allow relationship inference, individual behavior modeling, and group behavior analysis [4].

Privacy issues may in general prevent massive logging of audio, however in certain applications the benefits may outweigh privacy concerns. For example in a nursing home application the system may prevent fatal conflicts. Within a team of highly interacting professionals under critical conditions it may also be productive to allow group logging, e.g., to answer questions like 'who heard what?', or to allow a member of the group to 'rewind' and hear a particular utterance from the closest microphone.

Here we focus on the machine learning problem of resolving the interaction pattern within a set of audio LifeLogs. We consider a small simulated indoor environment with a few subjects. Our subjects enter into different configurations and our aim is to identify the resulting subgroups. Each subject produces speech and a personal microphone records a mixture of own speech and speech from subjects participating in the given subgroup (present in the same room). Our task is to identify the group structure. We phrase this as a sequence of ICA problems. In a given configuration we estimate which subset of subjects is present in a given personal recording, hence, was present in the

given subgroup. This is represented via the separation matrix: A non-zero element indicates that the particular speaker was present in a given microphone, while a zero element indicates that the speaker was absent.

Audio considerations: 1) Potential large distance between microphones can cause delays, we test the hypothesis that this can be handled by down sampling and instantaneous ICA. 2) We test the impact of adding white gaussian noise of different levels to the recordings.

2 Methodology

ICA is defined using the linear model $\mathbf{X} = \mathbf{AS}$, with $\mathbf{X}_{L\times N}$ being the N L-dimensional observations, $\mathbf{S}_{L\times N}$ being the sourcematrix, and $\mathbf{A}_{L\times L}$ the linear mixing. In ICA only $\mathbf{X}$ is known and $\mathbf{A}$ and $\mathbf{S}$ is found assuming statistical independence between the sources.

A number of different approaches have been proposed [9]. We focus on the maximum likelihood approach (ML) and use the inverse hyperbolic cosine distribution, $p(s) = \frac{1}{\pi \cosh(s)}$, for the sources, which is suitable for most super-gaussian signals. The negative log likelihood of the complete observation matrix, $\mathbf{X}$, given the separation matrix, $\mathbf{W}$, is,

$$\mathcal{L} = -\log \prod_n p(\mathbf{x}^n|\mathbf{W}) = -N \log|\det \mathbf{W}| + \sum_n \sum_i \log\cosh(s_i^n) + NL\log\pi.$$

This is used as an error function and is minimized using an algorithm based on the BFGS method [14]. The result is the maximum likelihood estimate of the separation matrix, and the likelihood of this solution is available.

Using ML for solving the ICA problem has the advantage that individual parameters can be trained separately or fixed to certain values. This will be used in the following where pruning of the parameters will be done. Because of problems with convergence and local maxima, we initialize $\mathbf{W}$ using FastICA[10] which is both fast and robust method of solving the ICA problem.

2.1 Pruning

We are interested in finding the structure in the separation matrix, meaning identifying the zero and non-zero parameters of the matrix. The approach we have chosen here is that of pruning, inspired by neural networks [1]. Pruning works by training a full solution, assessing which parameter is the least important, setting it to zero, retrain the solution, set another parameter to zero, and so forth. There are at least two questions that arise in this framework; how to assess the importance of a parameter and when to stop deleting parameters. These two question will be dealt with next.

2.1.1 Magnitude based pruning

Different measures have been suggested in the literature for assessing the importance of a parameter in the neural networks. In [13], OBD, OBS, ARD, and magnitude based pruning were compared for use in ICA. The conclusion was that magnitude based pruning was as good, if not better, than the three other measures, and being the simpler one as well, it became the obvious choice.

Magnitude based pruning is simply using the absolute value of each parameter as an indicator of the importance. This means the parameter with the smallest absolute value, is the one that is set to zero.

2.1.2 Bayesian information criterium

The second question when pruning was when to stop setting parameters to zero. In ICA there is a last point when the separation matrix becomes singular, but most likely it will give a better model stopping earlier. The likelihood of the model cannot be used directly because the likelihood will always be smaller for increasing sizes of nested models. Therefore other means are necessary. In [13] BIC, AIC, laplace approximation, and test set error was compared, and it was concluded that BIC was the best performing measure,

$$\text{BIC} = 2\mathcal{L} + L\log N.$$

The model with the lowest BIC value is chosen from the sequence of pruned models.

3 ICA experiments

To test the performance of the pruning, a set of experiments are run. Synthetic data is used. The sources are generated using the $p(s) = \frac{1}{\pi \cosh(s)}$ in one set of experiments and $N(0,1)^{1.5}$ in another. The separation matrix is 5×5, is generated using $N(0,4)$, and different numbers, $[5\,10\,15]$, of off-diagonal parameters are set to zero. The observations are then generated with $\mathbf{X} = \mathbf{W}^{-1}\mathbf{S}$. The results are compared to FastICA.

3.1 Performance measures

A number of different performance measures will be used. The mean square error of the extracted separation matrix, compared to the true matrix. The mean square error of the found sources, compared to the true sources. Because of the wish to identify zeros in the matrix a measure of false negatives, false positives, and the total number of misses is included. And then finally the measure from Amari [11], we call it the Amari error, is used as well,

$$E_A = \sum_i (\sum_j \frac{|p_{ij}|}{\max_k |p_{ik}|} - 1) + \sum_j (\sum_i \frac{|p_{ij}|}{\max_k |p_{kj}|} - 1),$$

with $\mathbf{P} = (p_{ij}) = \mathbf{A}\widehat{\mathbf{W}}$, $\widehat{\mathbf{W}}$ being the estimated separation matrix, and $\mathbf{A}$ is the true mixing matrix. This measure is scaling and permutation independent and exploits the fact that $\mathbf{P} = (p_{ij}) = \mathbf{AW}$ should be a permutation matrix. It is non-negative and zero if a perfect permutation matrix is found.

Because scaling and permutation is an issue in ICA, the matrices has been scaled and permuted to fit best with the known sources before the measures were calculated. Because there will always be some uncertainty in this process the measures are most reliable for sample sizes large enough to make the permutations of the matrices identifiable. The Amari error is unaffected by this.

3.2 Results

In figure 1 (A) is shown the average number of misses per experiment over a 1000 trials. Clearly, and not surprisingly, the misses are decreasing with increasing number of samples. It seems, that the more parameters that are zero the fewer misses. When looking at the false pos/neg counts, it is seen that more extra zeros are found for small number of zeros than extra non-zeros when many zeros are present. The missrate flattens out at a little under two, which is a bit high compared to the number of parameters (25). The parameters are generated using a gaussian distribution, so there will be non-zero parameters close to zero. In figure 1 (B), 0.5 has been added or subtracted to each parameter to avoid values close to zero. As can be seen the performance increases quite a bit and approaches zero, so it seems the error in (A) is because of non-zero parameters close to zero.

Not only do we get the structure of the separation matrix, but we actually get a better solution for both sources and separation matrix. This can be seen in figure 2 (A) and (B) where the mean square error of the parameters and the Amari error is shown. In the small sample sizes there are problems that could be explained by difficulties in the permutation of the sources, but in general the error is smaller for pruned ML over all sample sizes.

The pruned ML has the extra advantage of using the correct source distribution, but to show that it does not have to be strictly the same, an experiment has been run using another super-gaussian distribution; the exponentialized version of the gaussian distribution, $N(0,1)^{1.5}$. This is shown in figure 2 (C) and again the pruned ML increases accuracy.

The results were consistent over all the different accuracy measures even though there was not room to show them all.

4 Grouping LifeLogs

In this section we will use the described methods to identify subgroups in simulated LifeLog audio recordings. The setup simulates an environment in a company or an institution where employees wear a microphone, which could simply be a mp3 recorder or a PDA. If we imagine 5 recordings

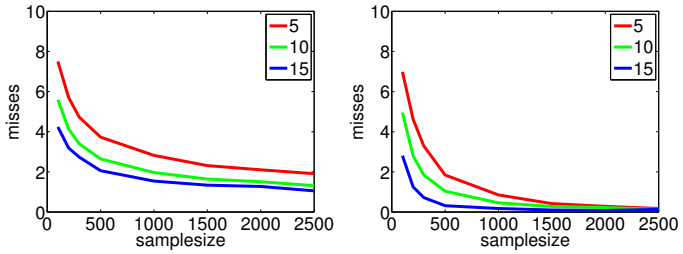


Figure 1: The number of misses (false negatives + false positives). (B) 0.5 has been added or subtracted from all elements in the separation matrix to avoid values close to zero.

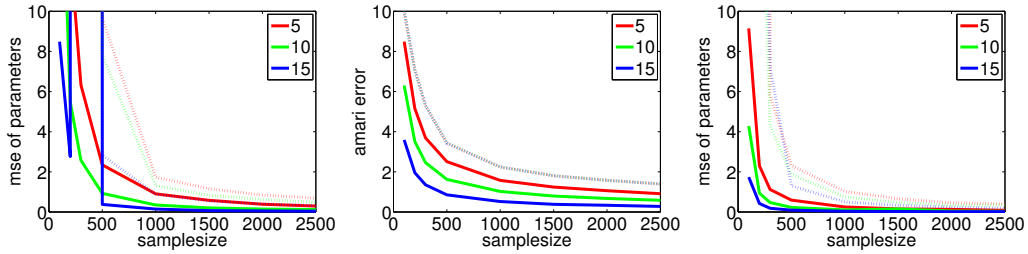


Figure 2: (A) Mean square error of the parameters. (B) The Amari error. (C) The mean square error of the parameters for source distributions using $N(0,1)^{1.5}$. All three shows the same trend of enhanced accuracy of pruned ML (full) versus FastICA (dotted). Red, green and blue are for 5, 10 and 15 zero parameters.

from 5 people. Four of them have two on two meetings and the last is by himself. The mixing matrix of such a problem would be a block diagonal matrix, for example like shown below,

$$\begin{pmatrix} 3 & 2 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 2 & 0 \\ 0 & 0 & 2 & 3 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{pmatrix}$$

If this structure could be recovered, the relation between the presence of each of sources could as well. The FastICA solution to a problem like this gives,

$$\begin{pmatrix} 2.6163 & 2.0525 & -0.0451 & -0.0513 & -0.0107 \\ 1.1340 & 1.9790 & 0.0222 & 0.0488 & 0.0114 \\ 0.0159 & 0.0080 & 2.2167 & 1.6409 & 0.4294 \\ -0.0132 & -0.0523 & 1.1840 & 1.9920 & -0.0174 \\ -0.0022 & -0.0017 & -0.8782 & -0.6429 & 0.9053 \end{pmatrix}$$

The structure can be seen in this matrix, but it is not clear, and exactly how many zeros are present is not apparent. Using the proposed method the following result is obtained,

$$\begin{pmatrix} 6.2496 & 4.8731 & 0 & 0 & 0 \\ 2.9893 & 5.1859 & 0 & 0 & 0 \\ 0 & 0 & 5.2500 & 3.8528 & 0 \\ 0 & 0 & 2.5972 & 4.2961 & 0 \\ 0 & 0 & 0 & 0 & 2.4036 \end{pmatrix}$$

Of course the scaling is not correct as is always the problem with ICA, but the correct number of zero parameters have been found and the structure is very clear. The sources has been permuted to fit.

4.1 Delays in recorded sound

A 1 kHz signal has a wavelength of ~ 0.3 m, meaning that distances should be less than this. For a sampling rate of 16 kHz the distance becomes very small, which is of course not the case for real settings. A solution is to use convolutive ICA which models this delay (and filtering) directly, but the problem becomes very complex. Another solution is to limit the frequency content of the sources by lowpass filtering and downsampling, and we will investigate this approach here.

A filtering of the signal is a linear sum of random variables, and according to the central limit theorem sum of random variables becomes more gaussian than the individual variables. ICA does not work for gaussian variables, so there is a limit to how much you can downsample the signal before the algorithm breaks down. Also information is lost during this process, so experiments will be run to find an appropriate downsampling factor.

4.2 Grouping recordings

The separation matrix is not the desired output, so a way of converting the separation matrix to subgroup structure is necessary. For the first matrix in equation 2 the masking matrix looks like this,

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{1}$$

Each recording works with a column in the matrix. A group in this matrix then is the same as a particular pattern of ones in a row. Recordings belonging to the same row pattern are grouped together. In the perfect case this gives a perfect grouping.

In the case of errors the approach must be extended a bit. This is done in the following way. The different row patterns are found. The number of occurrences of each pattern is found. Then, the pattern with most occurrences is chosen and all recordings activated for this pattern are assigned to the first group. Then, the second most occurrent row pattern is chosen and all the recordings activated in this are assigned to group two, excluding any recordings that has already been assigned. This is repeated until all recordings are assigned. An example where an error occurred in the masking matrix from equation 1, and where the sources have not been permuted correctly is shown below,

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

An error occured in the first row. Three row patterns are present $[0\,1\,0\,1\,1]$, $[1\,1\,1\,0\,0]$, $[0\,0\,0\,1\,1]$. The second pattern is the most occurring and recordings 1, 2 and 3 are assigned in this group. The two remaining patterns occurs equally often (once), but the error was corrected in the first assignments and the next pattern assigns recordings 4 and 5 to a new group, which completes the assignments. In this example the error did not influence the extracted groups.

4.3 Creation of recordings

The source data is taken from the TIMIT corpus of clean speech. Source 1 and 2 are interchangeable speaking or silent for periods of 3 seconds. Source 4 and 5 are created in the same way. Source 3 is a continuous source without silenced parts. The setup is two rooms, two speakers in each, source 1 and 2 are having a conversation in the first room and source 4 and 5 in the second room. The two rooms are connected via a corridor. Source 3 is first in room one, then goes to the corridor and then enters the second room. The situations are illustrated in figure 3 and the separation matrices are,

$$\begin{pmatrix} 3 & 2 & 1 & 0 & 0 \\ 2 & 3 & 1 & 0 & 0 \\ 1 & 1 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 2 \\ 0 & 0 & 0 & 2 & 3 \end{pmatrix} \begin{pmatrix} 3 & 2 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 2 \\ 0 & 0 & 0 & 2 & 3 \end{pmatrix} \begin{pmatrix} 3 & 2 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 1 & 1 \\ 0 & 0 & 1 & 3 & 2 \\ 0 & 0 & 1 & 2 & 3 \end{pmatrix} \tag{2}$$

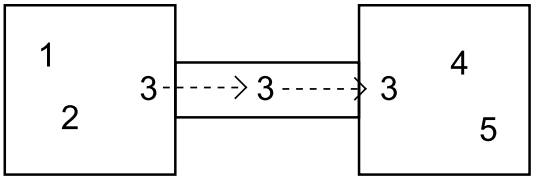


Figure 3: The setup of the LifeLog experiment. Recordings 1, 2, 4 and 5 are stationary. Recording 3 moves from the first room, to the corridor and then to the second room.

Table 1: Signal to noise ratio

SNR \ ds	80	160	240	320	400	480	560	640	720	800
0	138	140	136	151	153	142	145	159	139	150
-10	113	121	145	126	141	138	143	144	135	144
-20	74	101	100	109	133	131	116	126	124	132
-30	54	69	82	73	92	89	107	114	95	113
-40	23	48	72	65	77	78	78	81	80	78
-50	11	35	43	50	62	64	78	77	83	67
-60	13	40	50	56	54	51	59	64	68	62
-70	6	28	34	36	39	39	35	44	39	39
-80	0	4	3	6	10	16	12	19	24	32
-90	0	1	1	0	2	2	5	2	3	6
-100	0	0	1	0	0	1	1	1	3	2

The problem is assumed to be stationary in the three situations and the window size is one minute. 60 windows is used with 20 in each situation. The recordings in the TIMIT corpus are sampled at 16 kHz, so with the range of downsampling factors from 80 to 800, the sampling sizes range from 12000 to 1200.

Noise is added with different signal to noise ratios by adding white gaussian noise of differing amplitudes. Uniform random delays are added from each source to each recording in the range from zero to different maximum delays. The reported maximum delays are reported in meters, a maximum delay of 5 meters means that delays are generated in the range $[0; 5/344 \times 16000 = 232]$ samples, because the speed of sound is $344\frac{m}{s}$. Only integer delays are used.

4.4 Results

The experiment is run on 60 windows and 5 recordings. This means, that there are 300 groupings to be done. The misses are the total number of wrong labels over the 60 windows. If all windows were assigned to the same group it would give 160 misses. The first two rows are correctly assigned and the first 20 windows of recording 3 are also correct; $300 - 2 \cdot 60 - 20 = 160$. If all recordings were assigned to their own group it would also total 160. The first and last recording would be assigned correctly, and the middle 20 windows of recording 3 are right as well; $300 - 2 \cdot 60 - 20 = 160$.

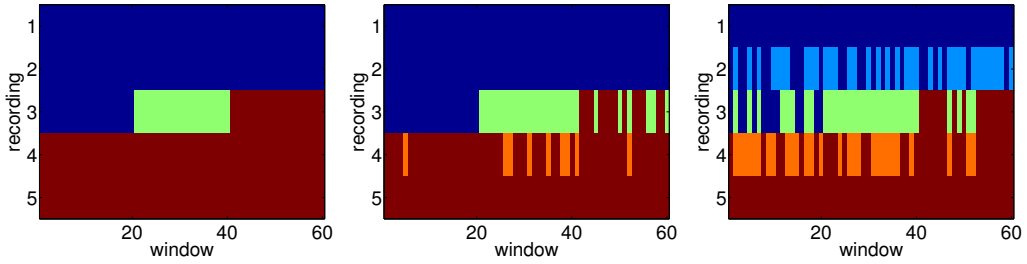
In table 1 is reported the misses for different levels of signal to noise ratio and downsampling factor. The algorithm seems very sensitive to noise and slightly more sensitive the higher the downsampling is. The border between usable and unusable data is around -70 db, which is quite low noise conditions. White gaussian noise is not a very likely noise condition in real recordings, but whether more realistic noise conditions will improve or worsen the results is hard to say.

In table 2 the misses as a function of delay and downsampling factor is shown. A triangular tendency can be seen with high error in the lower left part. This means that a high downsampling factor is good up to a certain point where too much is lost. Obviously the smaller the delays are the better, but it can be seen that the higher the downsampling factor the more delay can be accepted.

In figure 4 some examples from 2 are shown for different outcomes of the experiments. The first shows a perfect grouping. The second shows that some error (17 misses) can be accepted while still providing a clear picture of how the recordings have moved. The last figure shows an experiment with too much delay and the recordings almost get their own groups each time.

Table 2: Sound delay

m \ ds	80	160	240	320	400	480	560	640	720	800
0	0	0	0	0	0	0	2	7	6	12
1	10	0	0	0	0	0	2	2	7	12
2	6	0	1	0	1	0	2	3	6	10
3	9	14	0	5	0	0	2	2	6	10
4	22	10	15	5	6	4	6	3	6	10
5	13	29	0	10	16	2	1	3	5	8
6	45	13	1	35	76	36	1	17	6	13
7	18	13	36	22	18	14	13	5	9	18
8	24	19	23	38	25	24	2	0	3	15
9	45	66	26	42	40	60	28	14	7	13
10	15	57	24	23	24	49	36	35	9	17



(A) (B) (C)

Figure 4: Examples are taken from table 2. (A) perfect grouping with ds=80 and delay=0. (B) ds=400, delay=5 making 17 errors. The structure is still quite clear. (C) ds=400, delay=6, making 76 errors. The algorithm breaks down and the structure is only just visible.

5 Conclusion and discussion

We present a way of finding structure in the separation matrix of ICA. We do this using a pruning approach inspired by neural networks. By pruning the separation matrix fewer parameters are estimated and thus better estimates can be obtained with limited data available. This is shown for a number of different accuracy measures as compared to FastICA.

We use the proposed method to find structures in simulated LifeLogs recordings and present a way of organizing the recordings into groups of recordings present in the same setting. The proposed method is very simple and could possibly be extended and included information from the magnitudes of the estimated parameters.

We propose to use the downsampling of the sources to avoid the implications of delayed recordings and shows that, for clean speech in simulated mixings, quite good performance can be obtained for distances between the microphones and sources of up to around 5-6 meters.

In the simulation we add white gaussian noise to the recordings and find that the method is sensitive to this, and performance is degraded severely for signal to noise ratios higher than -70 dB. Whether this will be a problem for more realistic noise scenarios or true recording conditions remains to be seen.

References

[1] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, UK, 1996.

[2] Datong Chen, Jie Yang, Robert Malkin, and Howard D. Wactlar. Detecting social interactions of the elderly in a nursing home environment. *ACM Transactions on Multimedia Computing, Communications and Applications*, 3(1), February 2007.

[3] Defense Advanced Research Projects Agency (DARPA). Lifelog: Proposer information pamphlet, 2003.

[4] Nathan Eagle and Alex Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, V10(4):255–268, May 2006.

[5] Dan Ellis and Keansub Lee. Features for segmenting and classifying long-duration recordings of personal audio. In *ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing SAPA-04*, pages 1–6, 2004.

[6] Dan Ellis and Keansub Lee. Minimal-impact audio-based personal archives. In *First ACM workshop on Continuous Archiving and Recording of Personal Experiences CARPE-04*, pages 39–47, 2004.

[7] Eric Freeman and David Gelernter. Lifestreams: a storage model for personal data. *SIGMOD Rec.*, 25(1):80–86, March 1996.

[8] Jim Gemmell, Gordon Bell, Roger Lueder, Steven Drucker, and Curtis Wong. MyLifeBits: fulfilling the memex vision. In *Proceedings of the tenth ACM international conference on Multimedia (MM-02)*, pages 235–238, New York, December 1–6 2002. ACM Press.

[9] Aapo Hyvärinen. *Independent Component Analysis*. Wiley-Interscience, 1st edition, 2001.

[10] Aapo Hyvärinen. FastICA. http://www.cis.hut.fi/projects/ica/fastica/, 2006.

[11] Shun ichi Amari, Andrzej Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems 8*, pages 757–763, 1996.

[12] Nicky Kern, Bernt Schiele, and Albrecht Schmidt. Recognizing context for annotating a live life recording. *Personal Ubiquitous Comput.*, 11(4):251–263, 2007.

[13] Andreas Brinch Nielsen and Lars Kai Hansen. Structure learning by pruning in independent component analysis. *Neurocomputing*, 2007.

[14] Hans Bruun Nielsen. IMM optibox. http://www2.imm.dtu.dk/ hbn/immoptibox/, 2006.