FP7-ICT-2013-C TWO!EARS Project 618075

Deliverable 3.4

Progress report on feature selection and semantic labelling

WP3 *

November 27, 2015

* The TWO!EARS project (`http://www.twoears.eu`) has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 618075.

Project acronym:	TWO!EARS
Project full title:	Reading the world with TWO!EARS

Work packages:	WP3
Document number:	D3.4
Document title:	Progress report on feature selection and semantic labelling
Version:	1

Delivery date:	30th November 2015
Actual publication date:	30th November 2015
Dissemination level:	Public
Nature:	Report

Editors:	Guy Brown and Dorothea Kolossa
Author(s):	Ning Ma, Ivo Trowitzsch, Johannes Mohr, Klaus Obermayer, Christopher Schymura, Dorothea Kolossa, Thomas Walther, Hagen Wierstorf, Tobias May, Guy Brown, Patrick Danès, Michel Devy
Reviewer(s):	Jonas Braasch, Dorothea Kolossa, Bruno Gas, Klaus Obermayer

Contents

1 Executive summary

The Two!Ears project aims to develop an intelligent, active computational model of auditory perception and experience that operates in a multi-modal context. Ultimately, the system must identify the acoustic sources that are present in the environment and ascribe meaning to them. Progress on two key aspects of this goal within work package 3 (WP3) are presented in this report – selection of appropriate features for characterising sound sources, and approaches to semantic labelling.

The effectiveness of different acoustic features for the classification of acoustic sources is evaluated. It is found that an approach in which the Lasso technique is initially used, followed by construction of a linear classifier on the selected features, gives high performance while also drastically reducing the number of features to be computed.

Location and motion parameters are derived by a novel approach in which deep neural networks (DNNs) are used to map binaural features to the source azimuth. Furthermore, an approach is described for estimating the location and motion of acoustic sources that takes into account head movements, using a nonlinear dynamical system in which a control input is used to steer the head towards the desired orientation. Building on the feature selection work, approaches for learning and recognising source types are described that have very good overall classification performance for SNRs as low as 0 dB.

Labelling of visual objects using vision is discussed. Two approaches to audio-visual integration for speech recognition are presented, direct concatenation of audio and visual features, and joint recognition within a graphical model. Other approaches for integrating graphical models in the Two!Ears system are also presented. Segmentation is achieved by introducing graphical-model-based techniques from the field of computer vision. Graphical models also provide high-level analysis of acoustic scenes; semantic labels serve as observations for a Bayesian Network that describes specific properties of the acoustic scene; the remaining variables are then inferred via approximate inference techniques.

Preliminary work on scenario DASA-1 is reported, where the task is to identify the location of a female voice in the presence of four male-speech maskers. Using the approaches described above, it is shown that the five concurrent voices in this scenario can be localised and segmented. An approach for gender recognition was also described, which allows the system to discriminate the male and female voices. A scheme for using top-down feedback in the system is reported, which allows the Two!Ears system to exploit information about the source types present and the locations of masker sounds.

2 Introduction

2.1 Background to the TwoEars project

The Two!Ears project aims to develop an intelligent, active computational model of auditory perception and experience that operates in a multi-modal context. At the heart of the project is a software architecture based on a "blackboard system" that optimally fuses prior knowledge with the currently available sensor input, in order to find the best explanation of all available information.

Ultimately, the system must identify the acoustic sources that are present in the environment and ascribe meaning to them. Progress on two key aspects of this goal within work package 3 (WP3) are presented in this report – selection of appropriate features for characterising sound sources, and approaches to labelling. In the latter case, labels correspond both to source properties (e.g., the likely sound class, location in space, gender of a human voice) and semantics (e.g., is this source relevant to the current task of the system?). At the end of the report, we present work on a specific case study (scenario DASA-1) which will allow the complete Two!Ears system to be compared against human performance in a specific listening task.

2.2 Structure and function of the blackboard system

The current structure and function of the blackboard system is based on the architectural considerations that were presented in Deliverable D3.2. It is targeted as the front-end for a great variety of applications, providing an architecture that integrates experience formation and active behaviour from a set of individual functional modules. These modules can work on different levels of abstraction, independently from each other or in collaboration, in a bottom-up or top-down manner. A key feature of this system is its ability to evolve, so that easy modification, exchange and/or extension of modules can be achieved within a scalable architecture. The current implementation of the blackboard system is based on three main components:

Blackboard The blackboard holds the central data repository of the platform. It not only

stores current data, but keeps track of the history of this data in order to enable work on time series data.

Knowledge Sources knowledge sources (KSs) are modules that define their own functionality, to be executed in the organised frame of the system. They define which data they need for execution and which data they produce. The blackboard system provides the tools for requesting and storing this data, but does not care about the actual contents, while the KSs do not need to care about where and how data is stored.

Scheduler The scheduler is the component of the blackboard system that actually executes the KSs – but first, it schedules them, that is, it decides for the order in which KSs get executed. This order is rescheduled after every execution of a KS, since the conditions determining the order may have changed, or new KSs may be waiting for execution that are more urgent.

A general overview of the Two!Ears software architecture and the connections of the blackboard system to all other software modules is shown in Fig. 2.1. The blackboard system was released as part of the current Two!Ears auditory model, in conjunction with the corresponding documentation[1] of all its software components.

2.3 Structure of this report

The remainder of this report is structured as follows. Chapter 3 describes work on feature selection, which aims to find which features provided by the Two!Ears front-end are particularly informative for the classification of sound events. A classifier-based approach is taken. Chapter 4 concerns learning and semantic labelling; it describes how the system attaches meaningful labels to environmental sources, including their location and motion in space, the source type, and the identity of keywords that have been spoken (using both audio and visual information). The role of graphical models within the Two!Ears system is also discussed, both for scene segmentation and higher-level reasoning about the properties and semantics of environmental events. Chapter 5 reports preliminary work on a case study (scenario DASA-1) which involves the localisation of a female voice in the presence of 4 male-voice maskers. Work is reported on source segmentation, localisation, gender recognition and top-down feedback within the system. We conclude with a brief summary and discussion.

1 `http://twoears.aipa.tu-berlin.de/doc/1.0/blackboard/`

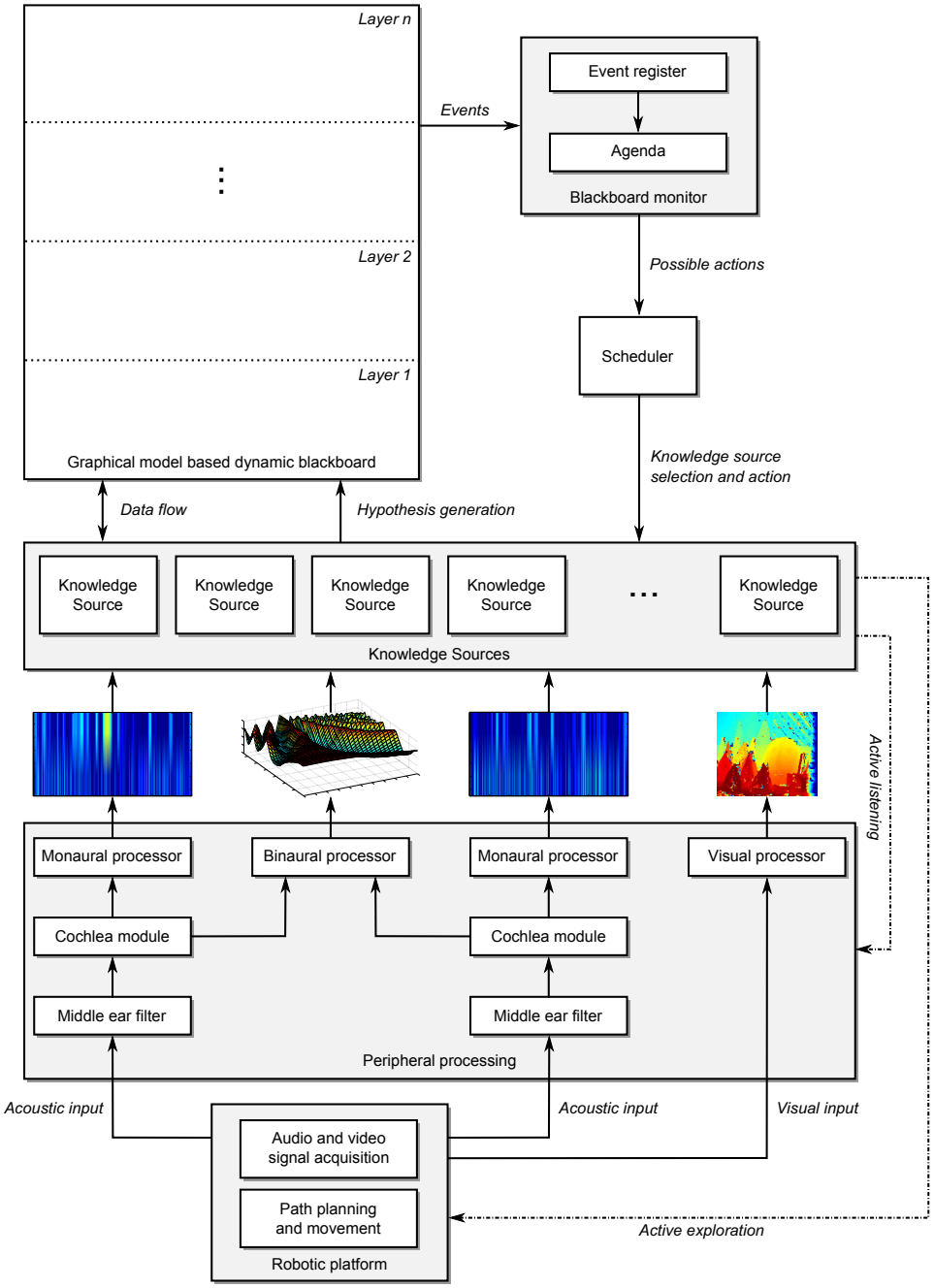


Figure 2.1: Overview of the general TWO!EARS software architecture.

3 Feature selection

3.1 Introduction

In this study we investigated the potential benefits of data-driven feature selection techniques for sound-type classification, and what kind of features provided by the Two!Ears front end are particularly informative for the subsequent classification by the identification knowledge sources. In order to do so, we used a classifier-based approach. On the one hand, features were selected using the classifier-based "Least Absolute Shrinkage and Selection Operator" (Lasso) technique. On the other hand, selected feature sets were evaluated with respect to the performance classifiers achieve which use the values of the selected features as inputs.

We selected four classes of target sounds from the NIGENS database and additionally constructed corresponding "non-target" sounds as well as "distractor" sounds using all other sound types. We then used the Binaural Simulator as part of our Auditory Machine Learning Training and Testing Pipeline to generate three sets of auditory scenes: (1) "dry" target sounds played from different spatial directions, (2) "dry" target sounds superimposed with ambient white noise of different strengths (SNR), and (3) "dry" target sounds overlaid with simultaneously played general distractor sounds of different strengths (SNR), where the azimuths of both were same as well as different.

While auditory classification will often be affected by additional acoustic disturbances such as reverberations, we restricted ourselves to studying robustness under the effects of source azimuth, ambient white noise, and point distractor sources in the current study. The methods for examining reverberant conditions in addition are not different from the ones used in this study.

All composed scenes were divided into time windows (blocks) of a few hundred milliseconds, and were subsequently preprocessed by the Auditory Front-end of the Two!Ears system to generate large sets of candidate features for further processing. We used a variety of feature sets as a starting point for the classifier-based feature selection procedures, among them monaural vs. binaural feature sets, sets using different frequency resolutions, sets using different block lengths, and sets of complex features derived from principal component and independent component analysis techniques.

We then constructed identification knowledge sources for sound-type classification using three machine learning schemes. The first scheme employed linear classifiers and support vector learning on all features. The second scheme employed the Lasso technique, i.e. a linear (logistic) regression model trained with a regularization term which penalizes a large number of input features. The second scheme thus selects informative features while simultaneously constructing the sound-type classifier. The third scheme implements a two-step procedure by combining schemes 1 and 2. First, features are selected using the Lasso technique. Second, a linear classifier is constructed using support vector learning on the selected features. This choice of schemes allows for an assessment of the potential benefits of data-driven feature selection. In addition it allows an assessment of robustness of results and the potential dependencies of classification performance on different training methods. The above mentioned machine learning schemes were then applied to the three sets of auditory scenes individually and in combination, and classifiers were constructed for the frame-based classification of sound-type.

In summary we find that the additional feature selection step employed by the learning scheme 3 usually does not improve average classification performance of a linear SVM compared to a setting where all elementary features are considered. The number of features to be calculated, however, can be drastically reduced with only minor to no compromises on classification performance. This also drastically cuts down on computation time necessary for training the classifiers and leads to shorter classification times.

Preprocessing using Independent Component Analysis leads to a much worse performance, and there are no clear benefits for training classifiers on feature sets generated by Principal Component Analysis compared to the elementary features. As both preprocessing techniques come with additional computational costs, we did not consider them further. The choice of the actual base feature sets does have an influence on classification performance, for instance there is a slight improvement in classification performance on average for the binaural feature set.

The performance of classifiers trained on a more diverse training set is slightly lower than the performance of classifiers trained on a specific condition - when evaluated on new data from the training condition(s). On the other hand, classifiers trained on a more diverse training set generalise better to conditions, which were not included in the training set. This is a general effect that is not specific to the base feature sets. Drops in performance under “cross-testing” can be due to differences in both the azimuth of the target sound source and the signal-to-noise ratio. Interestingly, a drop in performance across different azimuth angles of a target sound source can to a certain extent be compensated by using a monaural feature set which averages over both channels. On the diverse training set, the support vector machine using the full feature sets have a slight decrease in performance compared to the best Lasso and the two-stage lasso+SVM classifier. A more detailed analysis performed in section 3.5 showed that feature selection may improve performance

also for specialised knowledge sources performing on data from conditions they have seen for training, but that this comes at the expense of a less robust generalisation of these knowledge sources to data from conditions, which are different.

3.2 Data sets and preprocessing methods

The NIGENS database currently consists of 12 classes of everyday sounds (engine, crash, footsteps, piano, dog, phone, knock, fire, crying baby, alarm, female speech, and a general sound class). Each sound class consists of about 50 WAV-files, apart from the "general" class, which hosts 237 sounds chosen to exhibit as much variety (not included in the other sound classes) as possible. This special class is not intended to be a "positive" target of our classifiers, but provides counter examples to train against. All sound files have been manually annotated for on- and offsets of the target sound events.

We then created three data sets ("tasks") with simple auditory scenes using the Binaural Simulator plugged in as part of our Auditory ML Training and Testing Pipeline:

1. Scenes containing one sound source from the NIGENS database, located at four different azimuth angles: {0°, 45°, 90°, 180°}.

2. Scenes containing one sound source from the NIGENS database located at 0° azimuth, overlaid with ambient white noise at seven different signal-to-noise ratios: {∞, 20 dB, 10 dB, 5 dB, 0 dB, −10 dB, −20 dB}.

3. Scenes containing two sound sources (a "target" and a "distractor" source) from the NIGENS database which are played simultaneously. Target and distractor sound sources are located at three combinations of azimuth angles {0°&0°, −45°&45°, −90°&90°} and four values of signal-to-noise ratio {20 dB, 10 dB, 0 dB, −10 dB}.

Signal-to-noise ratios are measured/adjusted at the level of the "ear signals", i.e. after the separate binaural simulation of the sources.

Sound files were then decomposed into overlapping time blocks of 500 ms (also 200 ms and 1000 ms for one of the base feature sets) on which elementary candidate features (see below) were computed using the Two!Ears Auditory Front-end. The total number of positive examples per class were: 706 (female speech), 4368 (alarm), 3500 (crying baby), and 15105 (fire), with about 95000 negative examples (from all but the target class).

As a basis for feature generation, we used the following auditory representations provided by the Auditory Front-end:

ratemaps: auditory spectrograms that represent auditory nerve firing rates for each time frame (20 ms) and individual gammatone frequency channel (computed by smoothing the corresponding inner hair cell signal representation with a leaky integrator),

spectral features: 14 different statistics like flatness, kurtosis, etc., that summarise the spectral content of the ratemap for each time frame,

onset strengths: measured in decibel for each time frame and frequency channel, calculated by the frame-based increase in energy of the ratemap representation,

amplitude modulation spectrograms: each frequency channel of the inner hair cell representation is analysed by a bank of logarithmically-scaled modulation filters, so that for each time frame there are *number of frequency channels* $\times$ *number of modulation filters* values.

From this we constructed four different base feature sets which were then used as input for the machine learning methods:

Monaural: AFE representations (16-channel ratemaps, spectral features (built over 32 channels), 16-channel onset strengths maps, 8×9-channel amplitude modulation maps) are averaged over the left and right channel. Features are then calculated applying the following operations:

- For each block (for instance, 500 ms), compute the L-statistics[1] (L-mean, L-scale, L-skewness, L-kurtosis) of the representations over time.
- Additionally, build the first two deltas of the representation over time, corresponding to the discrete derivatives, and apply the L-statistics on these as well.

This finally amounts to 1082 dimensions per feature vector in this set.

Binaural: This set leaves out the averaging of the two ear channels done for the monaural set, otherwise exactly the same steps are taken (but on each channel separately), leading to a feature set of dimensionality 2164.

VarBlockLengths: To examine the effect of the length of the time windows (in the other sets fixed to 500 ms), here we basically triple the monaural set into one joint set which includes the features calculated over 200 ms, 500 ms, and 1000 ms, respectively.

1 L-statistics are given by L-moments, a sequence of statistics used to summarise the shape of a probability distribution (Hosking, 1990). L-statistics are shown to be more robust than conventional statistics, in particular with respect to the higher moments and when a small amount of data is available (David and Nagaraja, 2003, Ch. 9).

One vector then consists of 2982 features[2].

VarFrequencyResolutions: Additionally to the effect of block lengths, we wanted to have a look at the effect of frequency resolution. Thus we combined the base monaural feature set with a "high-resolution"-monaural feature set, which is based on a 48-channel ratemap and onset strength map, 64-channel based spectral features, and 24×9-channel amplitude modulation map. We have 4020-dimensioned feature vectors within this set.

Particularly for the latter two feature sets, the decision was to combine the three (block lengths), respectively two (frequency resolutions) qualities into one feature set rather than evaluating performance of three, respectively two separate feature sets, because this allows our feature selection method to actually choose between the different block lengths and frequency values.

3.3 Methods

3.3.1 Support vector machines

For classification, we used a linear C-Support Vector Machine (C-SVM). SVMs are classification models with associated learning algorithms that were derived in the context of statistical learning theory. Parameters are adjusted by maximizing the margin of a hyperplane separating the two classes, which can be related to a bound on the generalization performance of the classifier. If the training data is not linearly separable, so-called slack variables must be introduced that allow for violations of the margin. The sum of these slack variables serves as a penalty term and is weighted by the hyperparameter C. Here, we adjusted C via 4-fold cross-validation on the training set within the parameter set of $10^{-8}, 10^{-7}, \ldots, 10^{-2}$, using (eq. 3.1) as performance measure. The final classification performance was always evaluated on a held-out test set (cf. 3.3.5).

3.3.2 Lasso

Lasso is a linear logistic regression model with an L_1 penalty for the regression coefficients. This penalty forces many regression coefficients to be zero, leading to sparser models. Therefore, Lasso is a classification method with an embedded feature selection procedure. An important factor in determining the sparsity of the final model is the strength of the

2 It is not exactly $dim = 3 * dim(Monaural)$ because the onset strengths map was only calculated for eight channels here. This was missed to adjust to the same value as in the Monaural set.

L1 regularization term, which is controlled by the regularization parameter λ. We used two schemes for adjusting its value:

scheme fs1: We performed a 5-fold (or 7-fold) cross-validation on the training set for all 100 candidate values from the regularization path. We then chose the value with the best cross-validation performance (eq. 3.1).

scheme fs3: We performed a 5-fold (or 7-fold) cross-validation on the training set for all 100 candidate values from the regularization path. We then chose the highest value of λ whose cross-validation performance was greater than or equal to the difference between the maximum cross-validation performance over all λ and its standard deviation.

Application of criterion fs3 leads to features sets which are even sparser (less features) than features sets constructed using criterion fs1.

3.3.3 Lasso + SVM

We also employed a two-stage procedure, where Lasso was used to select two sets of features (fs1 and fs3) with non-zero coefficients for two different values of λ, which were determined via cross-validation on the training set (cf. 3.3.2).

These features were then handed as input features to a linear SVM, which was used for classification. As above, the hyper-parameter C of the SVM was adjusted by using 4-fold cross-validation on the training set.

3.3.4 Feature construction by principle component and independent component analysis

Classification performance depends on both type and number of features taken into account by a classifier. For example, correlations or statistical dependencies among features can degrade both model parameter estimates and overall performance. Furthermore, models using more features will typically fit training data better but face the danger of overfitting if the model gets too complex, i.e. generalising less well to new data.

We therefore studied the relation between model complexity (number of features) and classification performance for classifiers based on different features sets. We trained Lasso classifiers on the elementary features and compared them to classifiers trained on two alternative feature sets. We used Principal Component Analysis (PCA) to construct decorrelated features and Independent Component Analysis (ICA) to construct features which are statistically less dependent. For the latter we used fastICA with a log-cosh

contrast function. Both methods are commonly used as data preprocessing steps and automatically construct feature sets which have proven useful for many kinds of regression and classification problems.

For PCA, the extracted features are linear combinations of the original features and can be arranged such that they successively capture as much variance of the data set as possible. For independent components, however, no straightforward ordering exists with respect to their potential "informativeness".

In order to assess the utility of the different feature sets (elementary features, principal components, or independent components) and the dependence of performance on model complexity, we proceeded as follows: We used the regularisation parameter λ of the Lasso feature selection procedure to control model complexity and compared performance across feature sets. This allows us to judge which of these representations is particularly useful in combination with the classification methods (Lasso, SVM) used in this study.

3.3.5 Evaluation

We considered block-based sound classification tasks, where sounds of one type had to be classified against sounds of all other types. Data were split into training and test sets, where the training set was used to construct the classification model (i.e. the identification knowledge source) and the test set was used to evaluate the prediction performance. We considered two types of training schemes:

1. *single-conditional training*, where the identification knowledge source was trained on data taken from one type of auditory scene and one condition only, and

2. *multiconditional training*, where the identification knowledge source was trained on data taken across auditory scenes and conditions.

Performance was then measured either:

1. on test data chosen from the same combination of scenes and conditions as the training data (*iso-testing*) or

2. on test data chosen from a combination of scenes and conditions which was different from the combination used for training (*cross-testing*).

For classifiers constructed using the multiconditional training procedure "iso-testing" thus refers to the classification of new samples from a condition contained in the training set whereas "cross-testing" refers to the classification of new samples from conditions not contained in the training set.

Because the NIGENS database contains a roughly equal number of examples for each of the eleven target sound types and because each sound type was classified against the rest, there were many more negative than positive examples for each classification problem. For such imbalanced data sets, classification accuracy (correct classification rate) is not a valid performance measure, as an assignment of all data points to the larger class would already result in a good classification accuracy (without having learned any input-output relationship). A better performance measure is the so called "balanced accuracy", which is the arithmetic mean of sensitivity (true positives/size of the positive class) and specificity (true negatives/ size of the negative class). Because the used performance measure also highly influences the training process of classifiers through hyper-parameter search (for SVM: C, and for Lasso: λ), we constructed a modified version of balanced accuracy, which penalizes large differences between sensitivity and specificity:

$$\text{performance} = 1 - \sqrt{((1 - \text{sensitivity})^2 + (1 - \text{specificity})^2)/2)}. \tag{3.1}$$

For instance, in the absence of information about the true distribution of samples (and cost of errors) in later application of the trained models, we would prefer a classifier that shows a specificity of 0.8 and sensitivity of 0.8 over one that exhibits 0.6 versus 1.0. This performance measure in general is a bit more conservative than the standard balanced accuracy, in case of balanced specificity and sensitivity, it is equal to the standard balanced accuracy measure.

3.4 Results

3.4.1 Feature construction vs. feature selection

Here we studied the impact of model complexity (number of selected features) on classification performance. We used the monaural and binaural feature sets for the classification of target sounds in the presence of simultaneously played distractor sounds at different azimuths (target at 0° and distractor at 0° vs. target at −45° and distractor at +45° vs. target at 0° and distractor at 90°) for an SNR of 0 dB. We compared the performance obtained for the elementary features with the performance obtained for the features generated by principal and independent component analyses.

Figure 3.1 shows classification performance as a function of the number of input features for the 2×3 different feature sets. Model complexity was controlled via the regularisation parameter λ. Larger values of λ correspond to smaller numbers of features with non-zero weights, resulting in larger feature sets for smaller values of the regularisation parameter.

In particular for the classification of "alarm" sounds, the effect of overfitting is evident.

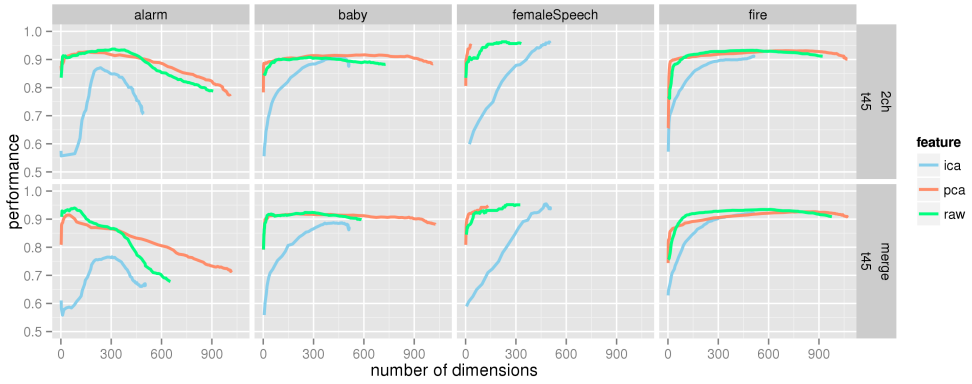


Figure 3.1: Classification performance of the Lasso-based classifier as a function of the number of input features ("dimensions"), i.e. the number of input features with non-zero coefficients, for the elementary features set ("raw") and the feature sets generated after preprocessing with PCA and ICA. Results are shown for the four classes "alarm", "crying baby", "female speech", and "fire", and for the monaural (2ch) and binaural (merge) feature sets.

Increasing model complexity by selecting more than an optimal number of features does not further improve performance but decreases generalisation performance. Principal components and elementary features result in similar peak performance values across classes and feature sets. The performance of independent components, however, is lower. This is not so surprising because the L1 regularisation used by the Lasso procedure selects input features which are mutually less dependent. Also the variability among different sound samples seems more useful for classification than independence of the representation.

We conclude that there are no clear benefits for training classifiers on the PCA or ICA generated feature sets. As both preprocessing techniques come with additional computational costs, we did not consider them further.

3.4.2 Performance-based evaluation of machine learning and feature selection schemes

First we asked, whether there is an overall difference in classification performance for the different learning schemes we have considered. The three schemes were:

- Scheme 1: Linear C support vector machines (SVM) trained on the whole set of elementary features (SVM-O, cf. section 3.3.1).
- Scheme 2: Logistic regression with L1 regularization (Lasso-fs1 & Lasso-fs3, cf. section 3.3.2).

- Scheme 3: A two-step procedure combining a Lasso feature selection step with a subsequent classification step using a linear C support vector machine. Lasso features were selected according to the criteria fs1 (SVM-fs1) and "fs-2" (SVM-fs3), see section 3.3.3.

Iso-testing

To evaluate the overall performance of the three classification schemes, we applied them to sound-type classification separately for all combinations of the four sound classes ("alarm", "baby", "female speech", "fire"), feature sets, scenarios, and conditions (see section 3.2). Classifiers were then evaluated on validation sets which contained data from the same combination of properties which were used for training ("iso-testing"). Summary performance results are plotted in Fig. 3.2.

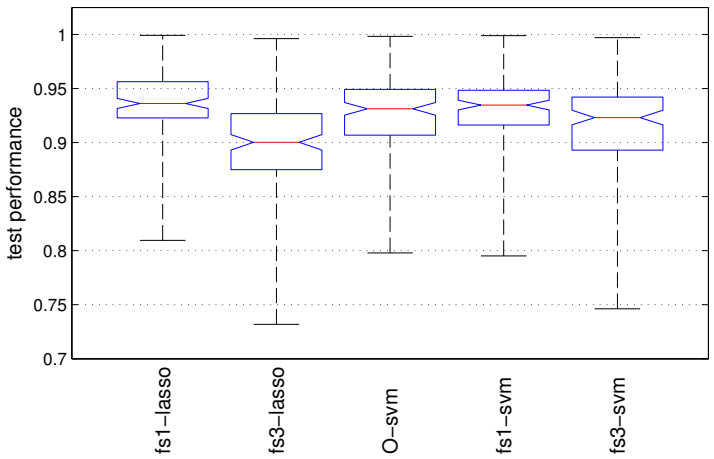
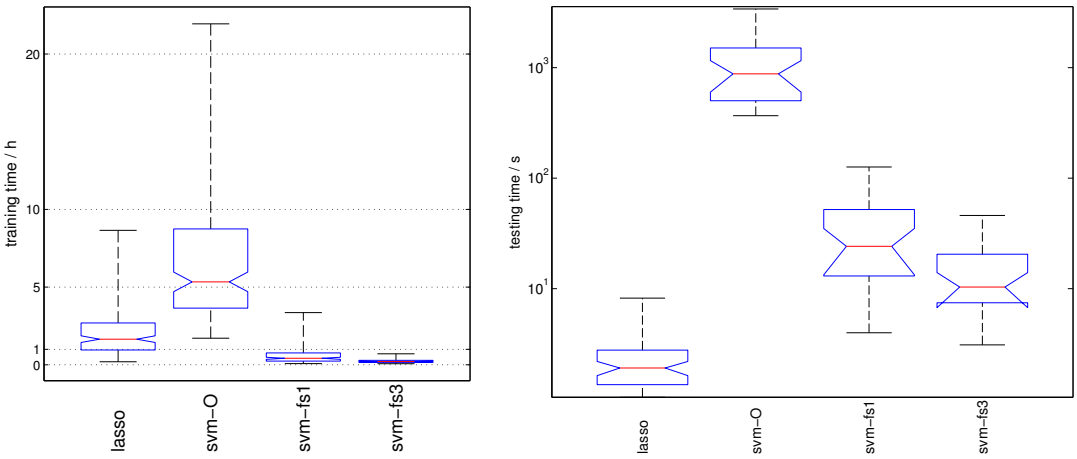


Figure 3.2: Classification performance of sound-type classifiers constructed using the three different learning schemes (see above) and the two feature selection criteria "fs1" and "fs3" (see section 3.3.2). Classifiers were constructed using single condition training and were evaluated using the iso-testing procedure (cf. 3.3.5). Each box plot summarizes the values obtained for the four classes, for the tasks 1 and 3, for all conditions, and for the base feature sets Monaural and Binaural.

These results demonstrate that feature selection using the stronger criterion fs3 resulted in a lower performance than feature selection using the criterion fs1. Since criterion fs3 always leads to feature sets which are smaller than the features sets derived from criterion fs1 (see also Fig. 3.12), underfitting may already set in, leading to a drop in classification performance. Otherwise, however, performance values are very similar. We conclude that the additional feature selection step employed by scheme 3 does not improve average classification performance of a linear SVM compared to a setting where all elementary



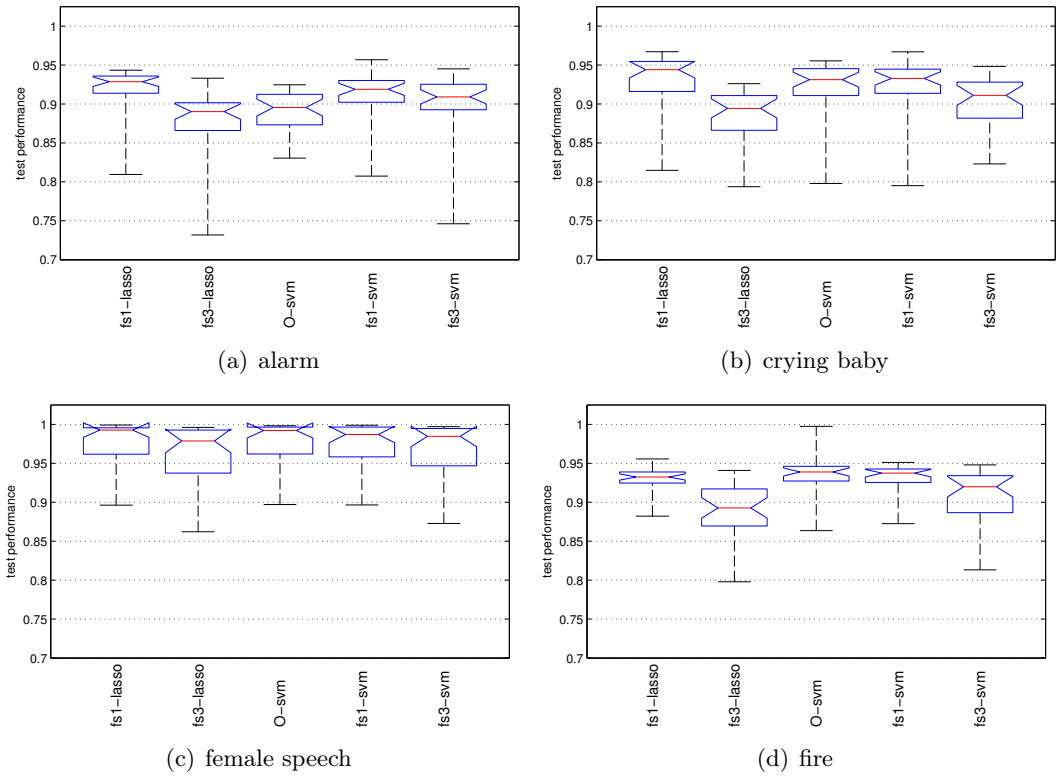
(a) training (b) testing

Figure 3.3: Training and testing times of sound-type classifiers constructed using the three different learning schemes and the two feature selection criteria "fs1" and "fs3". Classifiers were constructed using single condition training. Each box plot summarizes the times obtained for the four classes, for the tasks 1 and 3, for all conditions, and for the base feature sets "monaural" and "binaural". Training and testing was conducted with about 75000 vs 25000 samples. Please note that the training time-scale is in hours, and the testing time-scale in seconds, with a logarithmic axis.

features are considered, at least not for the tasks we have investigated. On the other hand, it does not reduce performance (for fs1), while using significantly fewer features than included in the base set, leading to an enormous drop in training and testing times, as can be seen in Fig. 3.3.

Figure 3.4 shows the overall performance of the three classification schemes separately for the four different classes. Consistent with the other findings (Fig.3.1 and further results below), we find class-specific differences in classification performance. Classifiers for female speech, for example, show a much better performance than classifiers for the other three sound classes. Slight differences in classification performance across the schemes Lasso-fs1, SVM-O and SVM-fs1 are too small to be interpreted. The pattern of relative performances across different classification schemes is very similar within classes.

We then compared the overall classification performance for the "monaural" feature set with the "binaural" feature set, where features were computed separately for the two input streams. On average, Fig. 3.5 shows that in tasks 1 and 3, there is a slight improvement in classification performance for the "binaural" feature set, except for Lasso-fs3.



(a) alarm (b) crying baby

(c) female speech (d) fire

Figure 3.4: Classification performance of sound-type classifiers constructed using the three different learning schemes and the two feature selection criteria fs1 and fs3. Box plots are shown separately for the four different classes "alarm", "crying baby", "female speech", and "fire". Otherwise, data was summarised as in Fig. 3.2.

Cross-testing

Since we are interested in *robust* (with respect to application under averse conditions) models, we now ask how classification performance changes if a knowledge source trained on data from a certain combination of sound class, feature set, task, and condition is evaluated on data produced by a different combination of those attributes ("cross-testing"). Here, we combine the cross-testing results across all the potential test combinations. We did not cross-test class and feature set (a "crying baby" classifier shall never classify fires and a classifier trained on one feature set cannot classify using any other feature set).

Figure 3.6 shows cross-testing performances for different classification schemes and for different feature sets. Comparing 3.6(a) with Figure 3.2, we see that cross-testing performance is lower than iso-testing performance for the different classification schemes. As

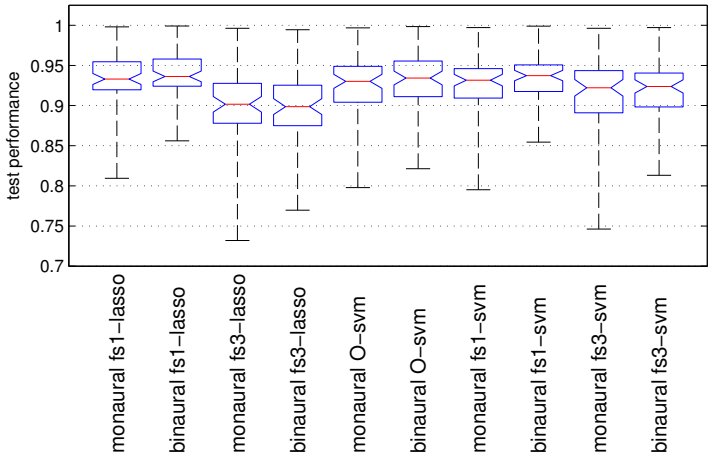


Figure 3.5: Classification performance of sound type classifiers constructed using the different learning schemes and feature selection criteria. Box plots are shown separately for the "monaural" and "binaural" feature sets. Otherwise, data was summarised as in Fig. 3.2.

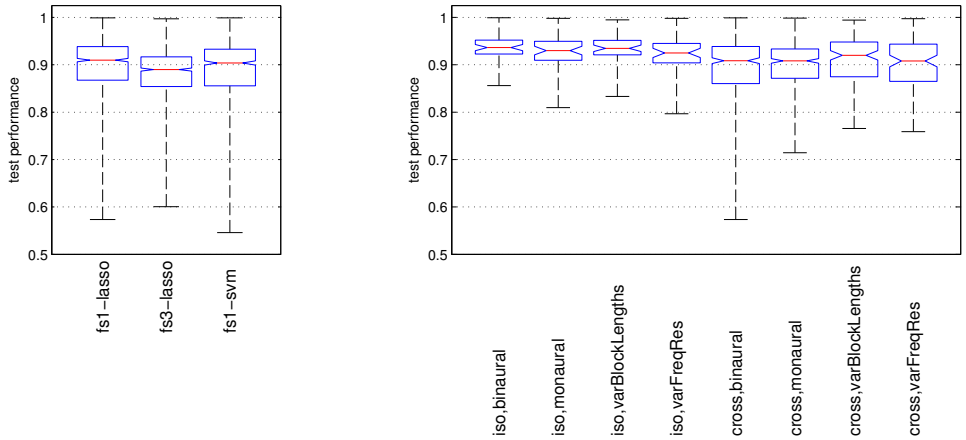


Figure 3.6: Cross-testing performance for different classifiers and feature sets. (a) Cross-testing results for the learning schemes Lasso-fs1 and Lasso-fs3 and the two stage training procedure SVM-fs1. Models and data from tasks 1 and 3. (b) Lasso classification performance in task 3 for different base feature sets. All classifiers were trained on single-condition data and evaluated either on data from the same ("iso-testing", four boxes on the left) or from the other conditions ("cross-testing", four boxes on the right). Data was summarised across all four classes and all conditions.

illustrated in Figs. 3.6(b) and 3.7, this is a general effect not specific to the base feature set used for classification.

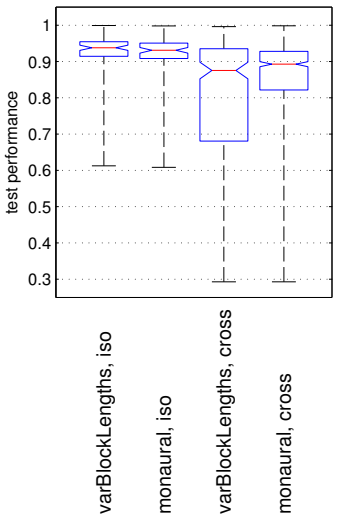


Figure 3.7: Classification performance of sound-type classifiers constructed by Lasso-fs1 and SVM-fs1. Box plots are shown separately for the "monaural" and "varBlockLengths" feature sets. Classifiers were trained on single-condition data and tested either on data from the same ("iso-testing") or the other conditions ("cross-testing"). Data was summarised across all four classes, tasks 2 and 3, and all conditions.

The drop in performance under cross-testing in task 3 is due to differences in both the azimuth of the target sound source and the SNR (see Fig. 3.8). While the binaural feature set performs a bit stronger under iso-testing, the decrease in classification performance observed for test stimuli at previously unseen azimuth values is higher than for the monaural feature set (which is less azimuth-dependent because of the averaging of values across channels, at the exense of a smaller iso-performance). Note that in this task, not only the azimuth of the target source is changed, but also the azimuth of the distractor source (for example from $(0°, 0°)$ to $(-90°, +90°)$).

Multi-conditional training

To examine whether the cross-testing shortcomings of classifiers trained on a specific condition can be overcome by training on a more diverse set of training samples, we created training sets with examples from multiple conditions and from more than one task. Not all conditions from these three tasks have been included into the multiconditional training set (e.g., from task 2, SNRs ∞, 0 and -10 have been included). Cross-testing for multiconditional models (similar to cross-testing for the single-condition case) always employs data from conditions not included in the training set (e.g., from task 2, SNR -20).

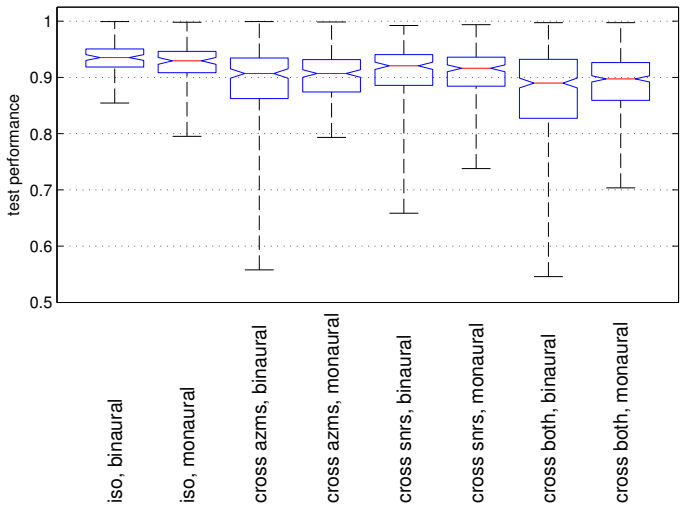
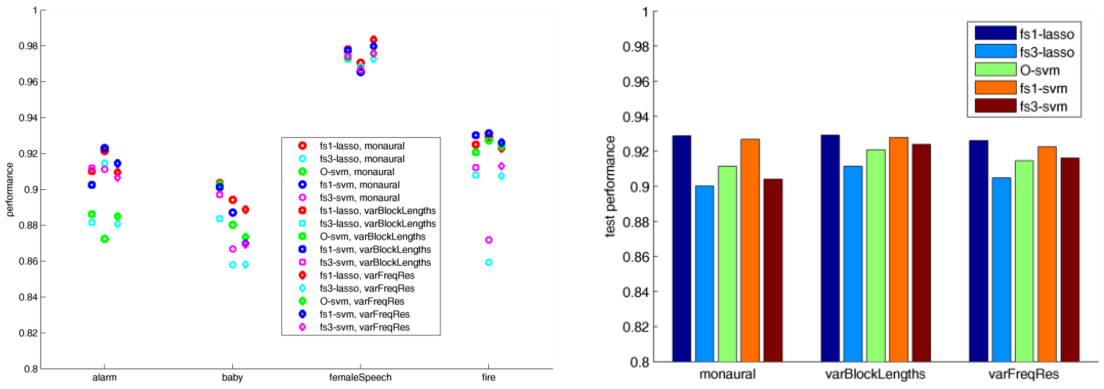


Figure 3.8: Classification performance using the monaural and binaural feature sets for task 3 under different cross-testing conditions. Classifiers were trained on single-condition for data using the Lasso-fs1 and SVM-fs1 learning schemes and evaluated using the iso- and three cross-testing procedures. Boxes labelled "cross azms" ("snrs") indicate cross-testing for varying azimuths (varying SNRs) of the sound sources while holding the SNR (azimuth) fixed. Boxes labelled "both" indicate cross-testing for both varying azimuths and SNR. Data was summarised across the two learning schemes, all four classes, and all conditions.



(a) per class (b) averaged over classes

Figure 3.9: Iso-testing performance of sound-type classifiers constructed using multiconditional training for the three different feature sets "monaural", "varBlockLengths", and "varFrequencyResolutions" (different symbols), and for the different learning schemes (indicated by color).

The iso-testing performance for multiple classifiers optimised using multiconditional training on different feature sets is shown in Fig. 3.9. Figure 3.9(a) shows that, consistent with single-

condition training, there are clear differences for the individual classes and classification schemes. But as can be seen in Fig. 3.9(b), the relative pattern of performance for different classification schemes is consistent across feature sets and with the findings from classifiers trained on single conditions: training according to criterion fs1 leads to better results than training to criterion fs3, and adding a subsequent SVM-classifier does not significantly improve performance. Note, however, one difference to the single-conditional training scheme performances: with the diverse multiconditional training set, the feature selection scheme fs1 improves the SVM, which has a performance drop using the whole feature sets.

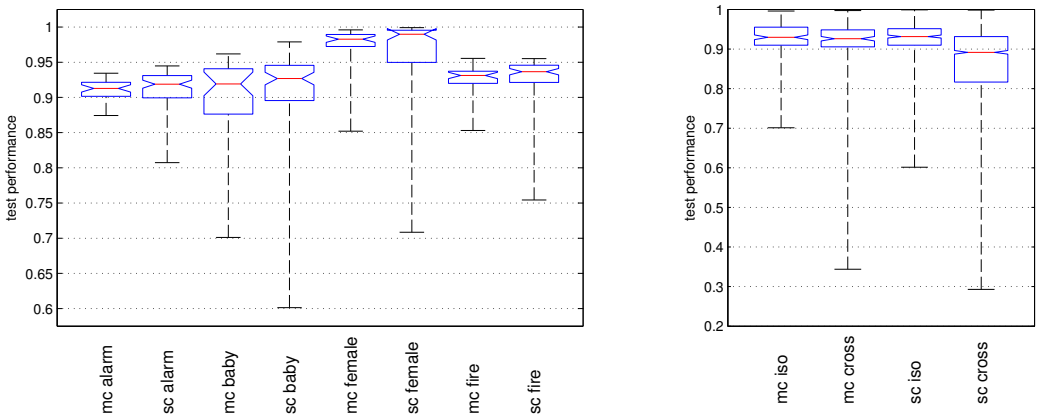
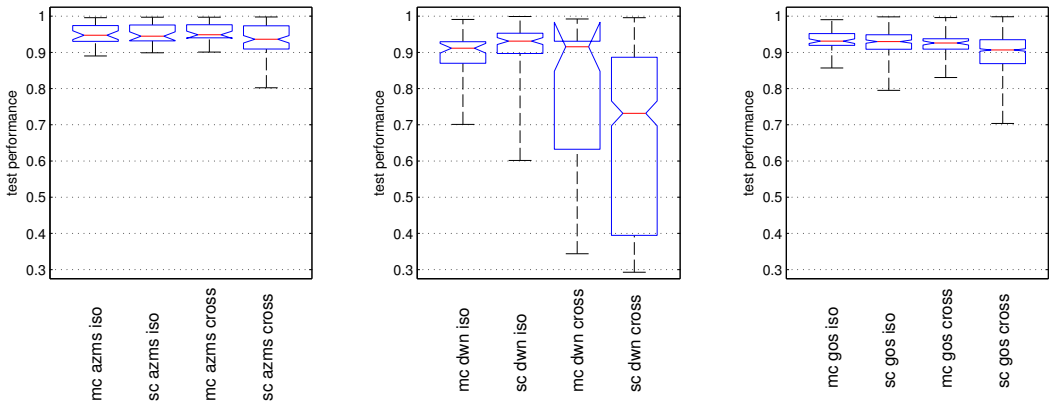


Figure 3.10: Performance of sound-type classifiers constructed with multiconditional training (mc) in comparison with the performance of classifiers constructed with single-condition training (sc). Performance was evaluated both using iso- (both panels) and cross-testing (right panel) procedure. Box-plots summarize data from classifiers trained with the Lasso-fs1 and SVM-fs1 procedures, using the "monaural", "varBlockLengths", and "varFrequencyResolutions" feature sets, and – for the single-conditional models – from all three tasks and all conditions.

Furthermore, Fig. 3.10 shows that the performance of classifiers trained on the multiconditional training set and tested in the iso-condition, is slightly lower than the performance of classifiers trained on the specific condition which matches the test condition. However, in line with the original motivation for multiconditional training, Fig. 3.10 indicates that the cross-testing drop in performance observed for the single-condition classifiers is much less severe for multiconditional trained models. This general trend is seen for different types of scene configurations as shown in Fig. 3.11.

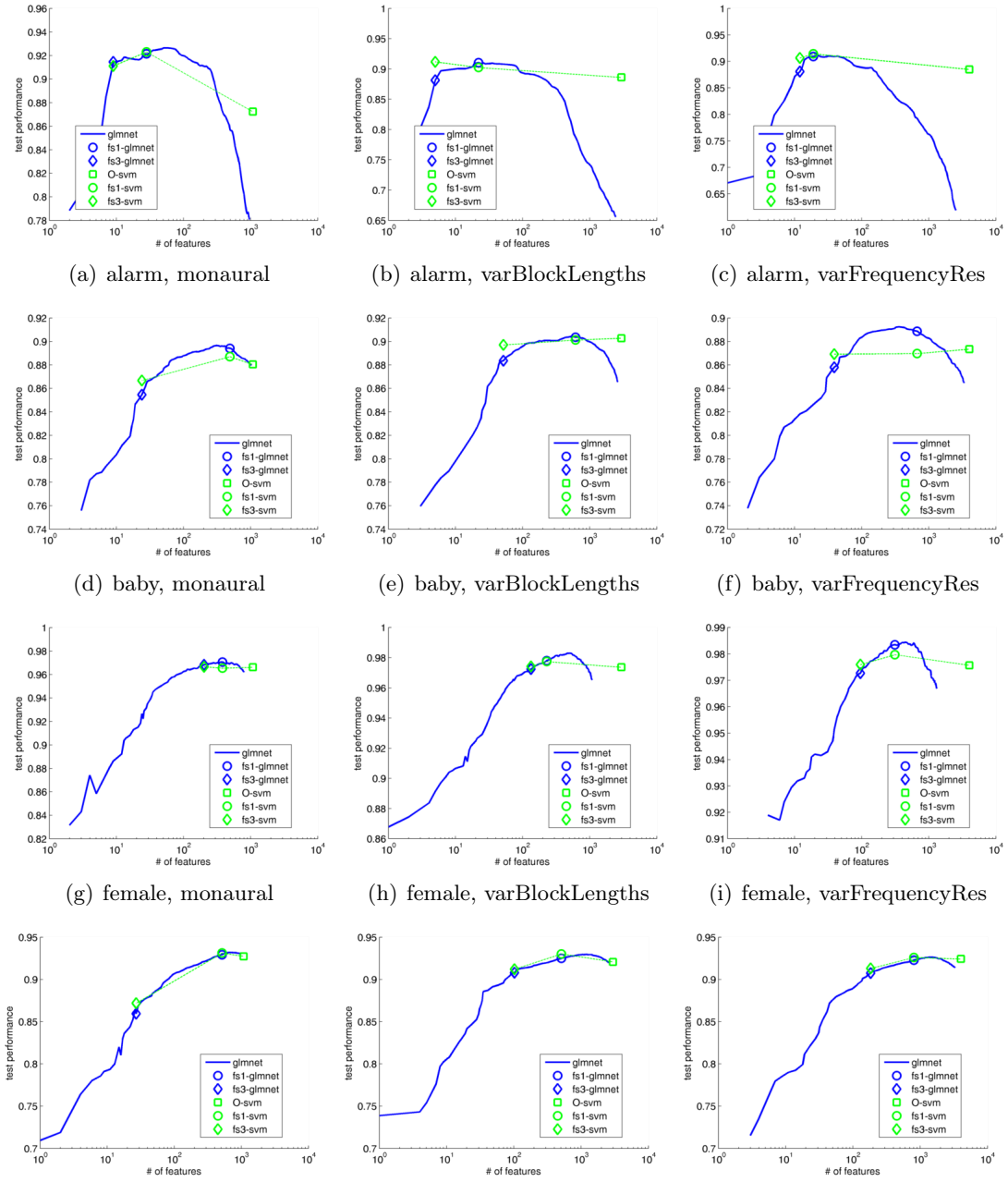


(a) task 1 (b) task 2 (c) task 3

Figure 3.11: Performance of sound-type classifiers constructed with multiconditional training (mc) in comparison with the performance of classifiers constructed with single-conditional training (sc). For all three tasks, the mc models were trained using the same training set, which includes conditions from all three tasks. The sc models have been trained on conditions from the respective tasks named in the subfigure. Boxes summarise data across results of the Lasso-fs1 and the two stage SVM-fs1 training procedures, the same feature sets as in Fig. 3.10, and all four classes.

Dependency of Performance on Number of Features

We next ask how classification performance depends on the number of features used for training a classifier. We varied the regularisation parameter λ of the Lasso method and obtained a set of features with non-zero weights for every value of λ. Figure 3.12 shows the classification performance on the validation set as a function of feature set size for the Lasso method (blue line). The classes "alarm" and "crying baby" show well defined performance peaks, while the classes "female speech" and "fire" show broader maxima. Low performance values indicate under- (towards small feature set size) and overfitting (towards large feature set size) effects. Markers appear at feature set sizes corresponding to the optimality criteria fs3 (leftmost symbols) and fs1 (central symbols), and at the feature set size corresponding to the full feature set (no selection, rightmost symbols). Depending on the class, the size of the optimal feature set can be very small (few percent for "alarm"), leading to a potentially large reduction in computation time for training and when implemented on the Two!Ears deployment system. Support vector machine classifiers (green symbols) are less prone to overfitting effects but otherwise do not exhibit clear performance advantages compared to the Lasso. Consistent with the results shown in Fig. 3.9 there is only a minor change in performance for SVM-based classifiers when trained on the full feature set. There are no qualitative differences between the different base feature sets with respect to the above



(a) alarm, monaural (b) alarm, varBlockLengths (c) alarm, varFrequencyRes

(d) baby, monaural (e) baby, varBlockLengths (f) baby, varFrequencyRes

(g) female, monaural (h) female, varBlockLengths (i) female, varFrequencyRes

(j) fire, monaural (k) fire, varBlockLengths (l) fire, varFrequencyRes

Figure 3.12: Classification performance on the validation set as a function of feature set size for the Lasso (blue line) and the SVM method, shown separately for the four classes and the three base feature sets "monaural", "varBlockLengths", and "varFrequencyResolutions". Classifiers were trained and tested on the multiconditional data in the iso-condition. The symbols denote the performances for the different learning schemes. The values for λ (fs1, fs3) were chosen according to the cross-validation performance which usually does not exactly coincide with the maximum of the validation performance on the test set.

conclusions.

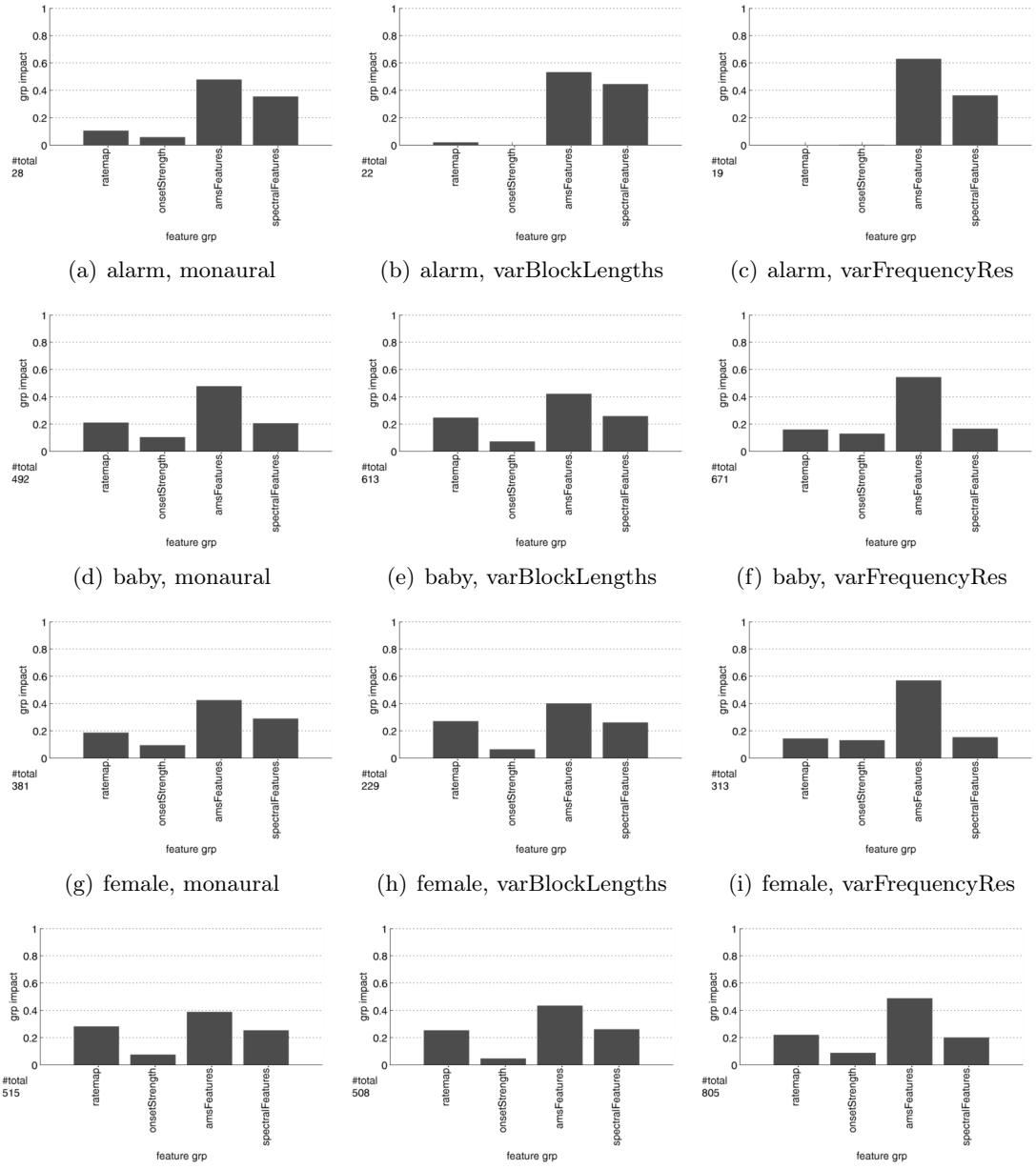
3.4.3 Feature profiles

The results of the previous section has shown, that the classification performances achieved with classifiers trained by the two-step machine learning scheme SVM-fs1 do not differ that much from the performances achieved when SVMs are trained on all available features on average - although the number of selected features may be as small as a few percent of the original feature set size. Selection scheme fs3 leads to even sparser sets, although it comes at the expense of a performance drop. The drop in performance, however, is small, and we conclude that most of the information needed for reliable classification is already present in a comparably small number of features which have to be computed for every block.

This holds for single- as well as multiconditional training. Since multiconditional training leads to classifiers, which are particularly robust against noise and distractor sounds, any selected features should be particularly well suited for classification in a cluttered environment. Fig. 3.13 shows the feature profiles for the feature sets selected by the Lasso method under the fs1 criterion for the multiconditional data sets. Amplitude modulation spectrogram features typically have the strongest effect (in terms of the sum of input features weighted by the absolute values of the logistic regression weights[3]) on classification results, followed by the spectral features, the ratemap, and the onset strengths features. There are quantitative differences between the profiles. However, there are no strong qualitative differences which would point towards a strong dependence of the profiles on the target sound-type, except for the total number of features selected (see also Fig. 3.12). As a trend, it seems that for the varFrequencyResolutions feature set which includes higher-resolution representations, the amplitude modulation features get an even higher share compared to the other features.

In order to explore the dependence on sound type further, we plotted the feature profiles for the four classes in more detail (see Fig. 3.14) for the monaural, multiconditional data set. Here we used the selection criterion fs3, because it leads to the most sparse feature sets, leaving only the most-informative features for the sound-type to be classified. Feature profiles now clearly vary across frequencies for the rate map, onset strength, and amplitude modulation features, across the subclasses of the spectral features, and across the modulation frequencies for the amplitude modulation features. Prominent for example are the different distributions across frequency for “alarm” and “female speech”, the almost

3 Data is scaled to zero mean and unit variance before training. We thus assume that the absolute values of the weights of the linear logistic regression model are representative for the “impact” of the corresponding features on classification.



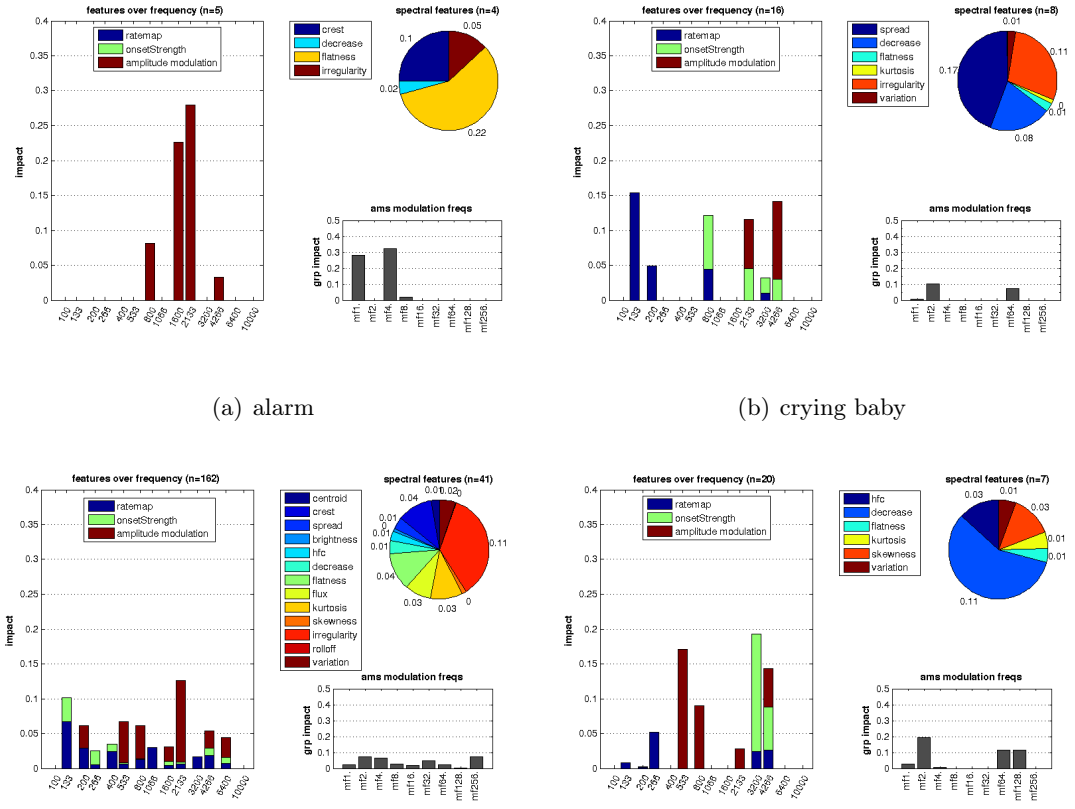
(a) alarm, monaural (b) alarm, varBlockLengths (c) alarm, varFrequencyRes

(d) baby, monaural (e) baby, varBlockLengths (f) baby, varFrequencyRes

(g) female, monaural (h) female, varBlockLengths (i) female, varFrequencyRes

(j) fire, monaural (k) fire, varBlockLengths (l) fire, varFrequencyRes

Figure 3.13: Feature profiles resulting from the application of the Lasso method (criterion fs1) to the multiconditional data set. The bar plots show the sum of weights (absolute values) of the linear logistic regression model - normalised to one - for the four different feature categories "ratemap", "onset strength", "amplitude modulation", and "spectral statistics". Separate profiles are shown for each class and for each of the three feature sets "monaural", "varBlockLengths", and "varFrequencyResolutions". The numbers in the lower left corners of the subfigures denote the total number of features with non-zero weights.



(a) alarm

(b) crying baby

(c) female speech

(d) fire

Figure 3.14: Feature profiles resulting from the application of the Lasso method (criterion fs3) to the multiconditional data for the Monaural feature set. Feature profiles are shown separately for the four target classes "alarm", "crying baby", "female speech", and "fire". The left panel in every subfigure shows the sum of input features weighted by the absolute values of the logistic regression weights ("impact") as a function of frequency for the "ratemap", "onset strength", and "amplitude modulation" features. The weighted sums for the different subclasses of the spectral features are plotted in the upper right panels. The panels on the lower left show a histogram of the weighted sums as a function of modulation frequency for the "amplitude modulation" features.

total dependence of "alarm" on the low-modulated amplitude modulation features, or the strong use of onset strengths only in very high frequency ranges for "fire". Similar variations occur for the other base feature sets (cf. Fig. 3.15 for the varFrequencyResolutions feature set).

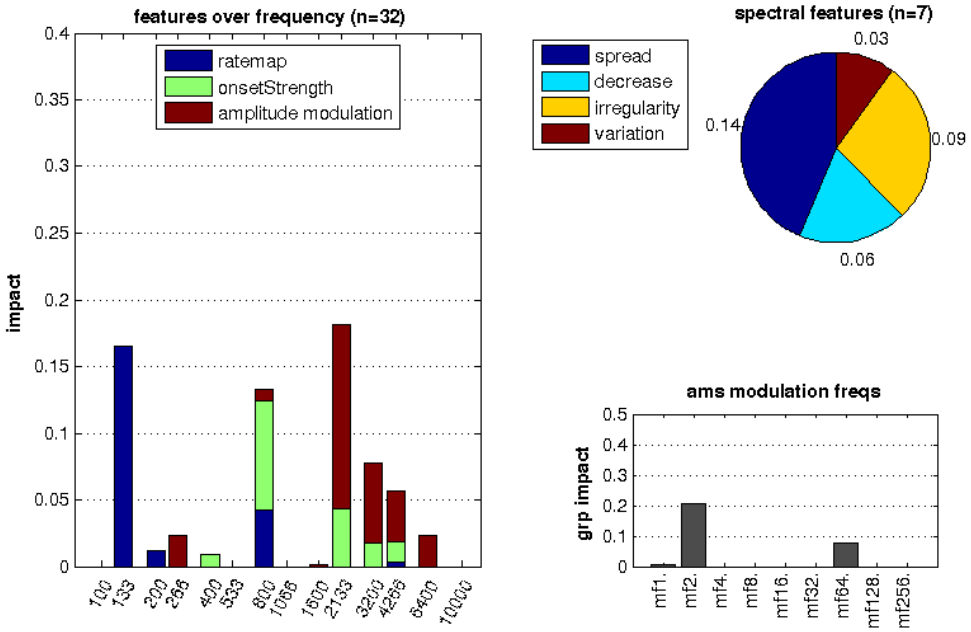
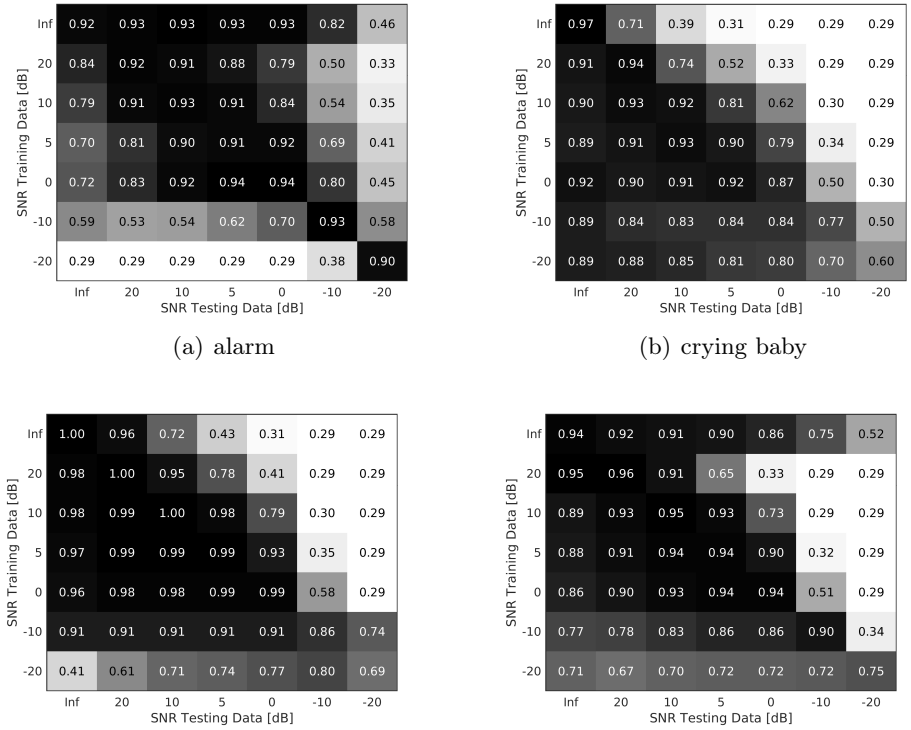


Figure 3.15: Same as Fig. 3.14(b) but for the base feature set varFrequencyResolutions.

3.5 Task 2: A detailed case study.

We conducted a more detailed analysis of the cross testing results obtained for task 2 (cf. section 3.2), where "dry" sounds emerging from sound sources at 0° azimuth were superimposed with ambient white noise of different strength (SNR) separately for the four classes "alarm", "crying baby", "female speech", and "fire".

The matrix plotted in Fig. 3.16 summarises the iso- and cross-testing results of support vector machine classifiers that were trained on the full feature set "monaural" for every SNR. In general, cross-testing leads to a significant loss of performance and the performance gradually becomes worse the more the SNR of the test data deviates from the SNR of the training data. For the two sound types "alarm" and "fire", classifiers trained in the no-noise condition generalize reasonably well to test data with SNRs down to 0dB where signal and noise have equal level (cross-test performance remains at 0.93 for "alarm" and 0.86 for "fire"). For the sound types "baby" and "female speech", on the other hand, the cross-test performance of the models trained at the no noise condition quickly decreases with decreasing SNR of the test data, dropping below 0.5 at SNR 10dB ("baby") and SNR



(a) alarm (b) crying baby

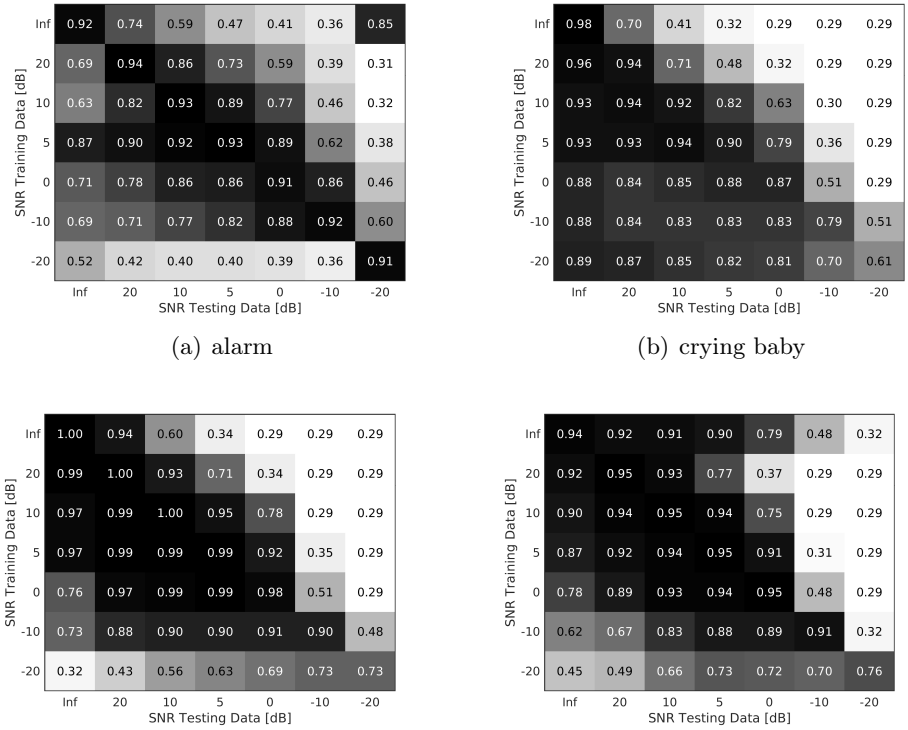
(c) female speech (d) fire

Figure 3.16: Iso- and cross-testing results of SVM-O classifiers that were trained on the monaural base feature set for every SNR for the four classes "alarm", "crying baby", "female speech", and "fire". Vertical and horizontal axes correspond to the SNRs used for training and test. Matrix entries denote the corresponding prediction performances (cf. eq. (3.1)). In addition, values are visualized by a grey level encoding from black (good performance) to white (bad performance).

5dB ("female speech"). These are also the classes that are generally harder to classify at high noise conditions, even when trained and tested under identical SNR. Iso-performance drops from 0.97 (no noise) to 0.6 (-20dB) for "crying baby" and from 1.00 (no noise) to 0.69 (-20dB) for "female speech".

Figure 3.17 shows the iso- and cross-testing results obtained with the Lasso-fs1 method for the same four classes and seven noise conditions shown in Fig. 3.16 and also for the feature set "monaural". The overall picture is very similar to the SVM case, however the performances of the Lasso-trained classifiers are significantly higher for the iso- (one-sided paired t-test over all sound types, P=0.034) but significantly lower for the cross-testing conditions (one-sided paired t-test over al sound types, P=0.001).

Moreover, we applied the Lasso method also to the feature set "varBlockLengths". The iso-



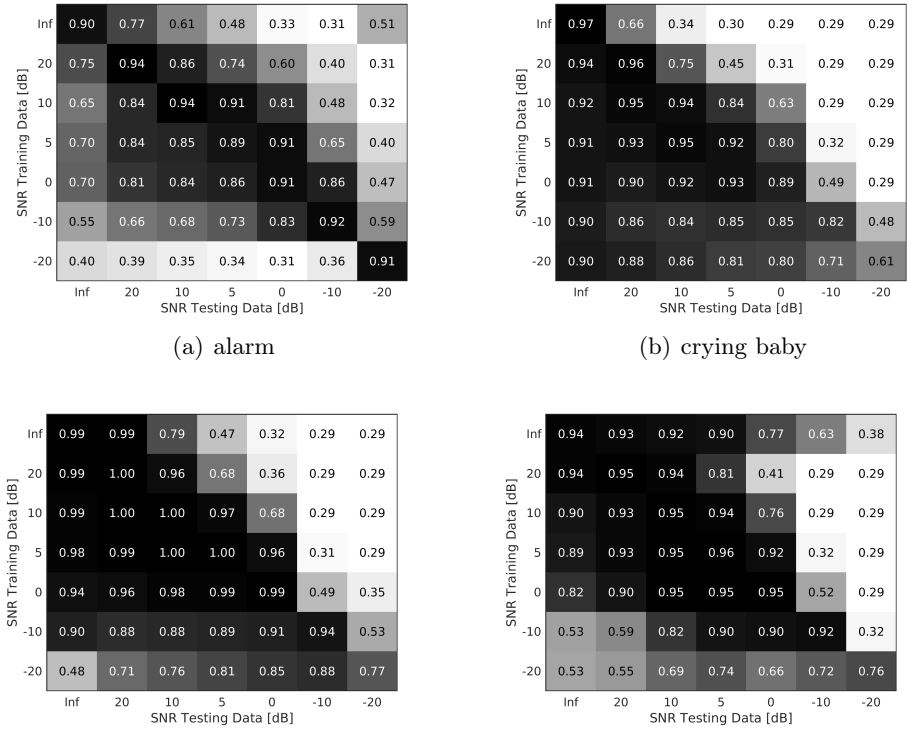
(a) alarm (b) crying baby

(c) female speech (d) fire

Figure 3.17: Iso- and cross-testing results of classifiers constructed by the Lasso-fs1 method for the "monaural" feature set for every SNR for the four classes "alarm", "crying baby", "female speech", and "fire". For details see caption of Fig. 3.16.

and cross-testing results are shown in Fig. 3.18. Although qualitatively similar, a statistical analysis shows that if trained and tested on identical conditions, the performance on the feature set " varBlockLengths" is significantly better than the performance on the feature set "monaural" (one-sided paired t-test over all sound types, P=0.008). No significant difference between these two feature sets was found for the cross-testing case.

Finally, we investigated the iso- and cross-testing performance of classifiers constructed using the two-step procedure SVM-fs1, where the Lasso method is used for selecting the features as the input into a linear SVM for classification. Results are shown in Fig. 3.19. For iso-testing, there is no significant difference in performance between the linear SVM with and without feature selection. For cross-testing, however, the SVM trained on all features ("SVM-O") yields significantly better performance (one-sided t-test, P=0.00002). These results suggest that feature selection via Lasso might lead to feature sets that are adapted to the particular noise condition and do not generalise well to test data from the other conditions. For a



(a) alarm

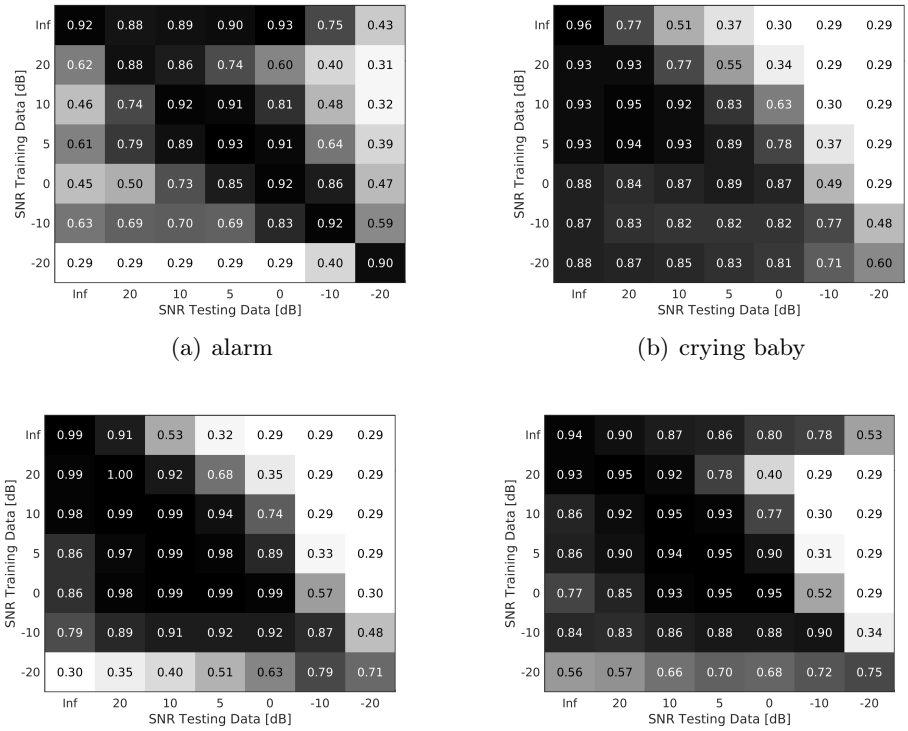
(b) crying baby

(c) female speech

(d) fire

Figure 3.18: Iso- and cross-testing results of classifiers constructed by the Lasso-fs1 method on the feature set "varBlockLengths" for every SNR for the four classes "alarm", "crying baby", "female speech", and "fire". For details see caption of Fig. 3.16.

particular noise condition, the number of necessary features can be drastically reduced without any significant decrease in classification performance, however the resulting model might not perform that well for different noise conditions.



(a) alarm (b) crying baby

(c) female speech (d) fire

Figure 3.19: Iso- and cross-testing results of classifiers constructed by the two-step method SVM-fs1 on the feature set monaural for every SNR for the four classes “alarm”, “crying baby”, “female speech”, and “fire”. For details see caption of Fig. 3.16.

4 Learning and semantic labelling

4.1 Location and motion parameters

Human listeners usually have little difficulty in localising multiple sound sources in reverberant environments, even though they must decode a complex acoustic mixture arriving at each ear Blauert (1997). In contrast, such adverse acoustic environments remain a challenging task for many machine localisation systems. The auditory system is able to exploit two main cues to determine the azimuth of a sound source in the horizontal plane: interaural time differences (ITDs) and interaural level differences (ILDs). Based on similar principles, binaural sound localisation systems typically localise sounds by estimating the ITD and ILD in a number of frequency bands, and employing statistical models such as Gaussian mixture models (GMMs) to map binaural cues to corresponding sound source azimuths. In this Section we present recent development in localisation of both static and moving sound sources, as well as active sound source localisation.

4.1.1 Sound localisation using deep neural networks

We present a novel machine-hearing system that exploits deep neural networks (DNNs) for robust localisation of multiple speakers in reverberant conditions (Ma *et al.*, 2015b). DNNs (Bengio, 2009) have recently been shown to be very effective classifiers, leading to superior performance in a number of speech recognition and acoustic signal processing tasks. Here, DNNs are used to map binaural features (obtained from a cross-correlogram) to the source azimuth. Two features, ITDs and ILDs, are typically used in binaural localisation systems Blauert (1997). ITD is estimated as the lag corresponding to the maximum in the cross-correlation function. In this study, instead of estimating the ITDs, we use the entire cross-correlation function as localisation features. This approach was motivated by two observations. First, computation of the ITD involves a peak-picking operation which may not be robust in the presence of noise. Second, there are systematic changes in the cross-correlation function with source azimuth (in particular, changes in the main peak with respect to its side peaks). Even in multi-source scenarios, the rich information can be exploited by a suitable classifier.

When sampled at 16 kHz, the cross-correlation function with a lag range of ± 1.1 ms

produced a 37-dimensional binaural feature space for each frequency channel. This was supplemented by the ILD, forming a final 38-dimensional (38D) feature vector. DNNs were then used to map the 38D binaural feature set to corresponding azimuth angles. A separate DNN was trained for each frequency channel. The DNN consists of an input layer, 8 hidden layers, and an output layer. The input layer contained 38 nodes and each node was assumed to be a Gaussian random variable with zero mean and unit variance. Therefore the 38D binaural feature input for each frequency channel was first Gaussian normalised, before being fed into the DNN. The hidden layers had sigmoid activation functions, and each layer contained 128 hidden nodes. The number of hidden nodes was heuristically selected as more hidden nodes add more computation and did not improve localisation accuracy in this study. The output layer contained 72 nodes corresponding to the 72 azimuth angles in the full 360 deg azimuth range (5 deg steps) considered in this study. The "softmax" activation function was applied at the output layer.

The neural net was initialised with a single hidden layer, and the number of hidden layers was gradually increased in later training phases. In each training phase, mini-batch gradient descent with a batch size of 256 was used, including a momentum term with the momentum rate set to 0.5. The initial learning rate was set to 0.05, which gradually decreased to 0.001 after 10 epochs. After the learning rate decreased to 0.001, it was held constant for a further 5 epochs. At the end of each training phase, an extra hidden layer was added before the output layer, and this training phase was repeated until the desired number of hidden layers was reached (8 hidden layers in this study).

Given the observed feature set $\vec{x}_{t,f}$ at time frame t and frequency channel f, the 72 "softmax" output values from the DNN for frequency channel f were considered as posterior probabilities $\mathcal{P}(k|\vec{x}_{t,f})$, where k is the azimuth angle and $\sum_k \mathcal{P}(k|\vec{x}_{t,f}) = 1$. The posteriors were then integrated across frequency to yield the probability of azimuth k, given features of the entire frequency range at time t

$$\mathcal{P}(k|\vec{x}_t) = \frac{\prod_f \mathcal{P}(k|\vec{x}_{t,f})}{\sum_k \prod_f \mathcal{P}(k|\vec{x}_{t,f})}. \tag{4.1}$$

Sound localisation was performed for a signal chunk consisting of T time frames. Therefore the frame posteriors were further averaged across time to produce a posterior distribution $\mathcal{P}(k)$ of sound source activity

$$\mathcal{P}(k) = \frac{1}{T} \sum_{t}^{t+T-1} \mathcal{P}(k|\vec{x}_t). \tag{4.2}$$

The target location was given by the azimuth k that maximises $\mathcal{P}(k)$

$$\hat{k} = \arg\max_k \mathcal{P}(k) \tag{4.3}$$

Table 4.1: Gross accuracy in % for various sets of binaural room impulse responses (BRIRs) when localising one, two and three competing speakers.

System	Surrey Room A			Surrey Room B		
	1-spk	2-spk	3-spk	1-spk	2-spk	3-spk
GMM	92.6	86.3	72.3	87.5	77.6	66.5
+ Head Movement	99.9	92.1	76.4	99.5	86.4	71.4
DNN – Full	99.9	88.7	78.5	94.1	81.5	74.1
+ Head Movement	99.8	97.1	86.0	99.9	94.9	83.8

Table 4.1 lists gross localisation accuracy rates of the DNN-based localisation system and a GMM-based system May *et al.* (2015) evaluated for various sets of BRIRs in the Surrey database Hummersone *et al.* (2010). Two BRIRs were used here and Room B contains a larger amount of reverberation. We also evaluated the performance when head movements were exploited as previously reported by Ma *et al.* (2015c), May *et al.* (2015).

The overall localisation accuracy of the full DNN system consistently outperformed the GMM-based system across all the testing conditions. The improvement was particularly pronounced in the single-speaker localisation task, with the DNN localisation accuracy approaching 100% in Room A. Across all speaker conditions the largest benefits were observed in Room B, where the direct-to-reverberant ratio is the lowest.

4.1.2 Estimation of motion parameters for moving sound sources

A framework for estimating the angular location and circular motion parameters involving head movements has been proposed in Schymura *et al.* (2015). It is based on a generic nonlinear dynamical system representation

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, u_k) + \boldsymbol{v}_k \tag{4.4}$$

$$\boldsymbol{y}_k = \boldsymbol{g}(\boldsymbol{x}_k) + \boldsymbol{w}_k\,, \tag{4.5}$$

where $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ denote the hidden state and the observation vectors at time frame k. The control input u_k is used to steer the head towards the desired orientation. $\boldsymbol{f}(\,\cdot\,)$ and $\boldsymbol{g}(\,\cdot\,)$ are nonlinear functions describing the model dynamics and the observations, namely interaural time differences (ITDs) and interaural level differences (ILDs), respectively. $\boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$ and $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$ are zero-mean, Gaussian distributed noise vectors, with covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$.

The model dynamics in Eq. (4.4) are represented by the 3-dimensional state vector

$$\boldsymbol{x}_k = [\phi_k \;\; \dot{\phi}_k \;\; \psi_k]^T$$

including the source position ϕ_k, the angular source velocity $\dot{\phi}_k$ and the head orientation ψ_k. The process equations of the former two can be described by

$$\begin{aligned}\phi_{k+1} &= \phi_k + T\dot{\phi}_k + v_{\phi,k}, \quad v_{\phi,k} \sim \mathcal{N}(0, \sigma_\phi^2)\\ \dot{\phi}_{k+1} &= \dot{\phi}_k + v_{\dot{\phi},k}, \quad v_{\dot{\phi},k} \sim \mathcal{N}(0, \sigma_{\dot{\phi}}^2),\end{aligned}$$

where T denotes the time between two consecutive measurements in seconds. σ_ϕ^2 and $\sigma_{\dot{\phi}}^2$ are the variances of the noise terms. The process equation of the look direction is represented as

$$\psi_{k+1} = \mathrm{sat}\Big(\psi_k + T\dot{\psi}_{\max}\,\mathrm{sat}(u_k,\, u_{\max}),\, \psi_{\max}\Big) + v_{\psi,k},\;\; v_{\psi,k} \sim \mathcal{N}(0, \sigma_\psi^2), \tag{4.6}$$

where $\dot{\psi}_{\max}$ is the maximum angular velocity for the head rotation in radians per second, which is assumed to be constant. In order to model physical constraints of the maximum head displacement and restricted control inputs, two saturation functions $\mathrm{sat}(x) = \min(|x|, x_{\max}) \cdot \mathrm{sgn}(x)$ are introduced in Eq. (4.6), where $\psi_{\max}$ is the maximum rotational angle and $u_{\max}$ is the control input limit.

The measurement equation (4.5) is based on a spherical head model introduced by Brungart and Rabinowitz (1999) and Algazi *et al.* (2001), which generates ITDs $\tau_m(\Delta\phi_k)$ and ILDs $\delta_m(\Delta\phi_k)$ based on the relative angle between the look direction of the head and the current estimate of the source position $\Delta\phi_k = \phi_k - \psi_k$, within a combined measurement vector

$$\boldsymbol{y}_k = \Big[\tau_1(\Delta\phi_k),\, \ldots,\, \tau_M(\Delta\phi_k),\, \delta_1(\Delta\phi_k),\, \ldots,\, \delta_M(\Delta\phi_k)\Big]^T,$$

where $m = 1, \ldots, M$ is the channel index of the auditory filterbank. A detailed description of both the process and measurement model with their parameters is presented in Schymura *et al.* (2015).

The proposed framework is capable of tracking the position and angular velocity of a moving sound source over time. Furthermore, the structure of process equation (4.4) implicitly allows for eliminating front-back ambiguities if head movements are applied. This is achieved by estimating the hidden state with an Unscented Kalman Filter (UKF). An example of the estimations generated by the model is depicted in Fig. 4.1.

Further extensions of the model will focus on including acceleration and distance estimation into the framework. This will ultimately allow for building probabilistic maps of potential source positions and trajectories within a head-centered coordinate system.

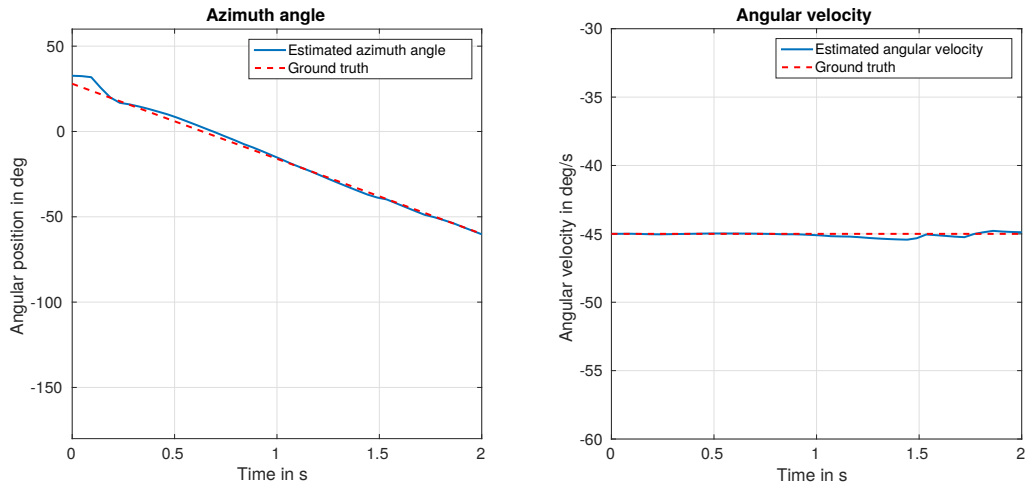


Figure 4.1: Example of source location and angular velocity estimation for a speech source in anechoic conditions.

4.1.3 Active localization

A binaural "active" sound source localization strategy has also been implemented. It is situated at the sensorimotor level of the model, and thus involves no cognitive nor decisional process. It consists in the joint processing and/or the interweave of binaural sensing and motion so as to disambiguate front from back and recover source range.

Its input is the sensorimotor flow, *i.e.* the binaural audio stream and the translation and rotation commands of the binaural head. A cascade of two processors, which can be viewed as experts of the model, process this information so as to get spatial information on the source. The first one consists in extracting directional cues on the basis of the short-term analysis of the channel-time-frequency decomposition of the audio signals. The second one assimilates these data over time and combines them with the motor commands so as to get a probabilistic description of the full set of variables characterizing the head-to-source relative situation. On the basis of these two knowledge sources, a separate expert also computes the one-step-ahead velocities to be applied to the binaural head in order to improve the quality of the localization. So, it enables the cognitive layer to trigger a sensorimotor feedback control, which sends reflexive motor commands to the head computed from the information extracted from the sensorimotor flow on the relative situation of the source, and taking account of other criteria (*e.g.*, limitations on motor commands).

This last sensorimotor feedback is the topic of Chapter 2.8 in Deliverable 4.2. As an introduction to the first two stages as well as references are also included for self-containedness, the reader is refered to that chapter.

4.2 Learning and recognising source types

4.2.1 Introduction

Below we summarize our investigations into how well machine learning procedures perform when applied to the problem of classifying everyday sounds in simple auditory scenes. We considered three different learning and classification schemes: (1) Lasso, a classification method with embedded feature selection, (2) a two-stage learning scheme, where the feature selection step using the lasso is followed by a classification step using a linear C support vector machine, and (3) a learning scheme which implements a newly developed probabilistic model. The latter model directly predicts class probabilities and allows the assessment of the confidence in a classification result in a way which is better grounded in theory than the methods typically used in connection with support vector learning.

We used sound examples from the 11 sound classes which are currently included in the NIGENS database. We then trained and evaluated identification knowledge sources using the above machine learning schemes on "dry" sounds (for scheme 3) as well as data obtained from the three simple auditory scenes that are described in chapter 3, section 3.2 (for schemes 1 and 2):

1. "Dry" target sounds played from different spatial directions,
2. "dry" target sounds superimposed with ambient white noise of different strength, i.e. varying signal-to-noise ratio (SNR), and
3. 'dry' target sounds overlaid with simultaneously played distractor sounds of different strengths (SNR), where the azimuths of both were the same as well as different.

All auditory scenes were divided into time-windows of 500ms, and were created and preprocessed by the Binaural Simulator and Auditory Front-end of the Two!Ears system (integrated into our Auditory Machine-Learning Training and Testing Pipeline, see section 4.2.4) to generate large sets of candidate features for further processing.

We then evaluated the classification results in order to answer the following questions:

1. What classification performance can be obtained for the different sound types?
2. Can performance be improved by creating specialized identification knowledge sources for particular conditions, for example for particular values of the signal-to-noise ratio characterizing an auditory scene?
3. How does performance change if these specialized knowledge sources are applied to data taken from conditions that are different from the ones used for training?

4. How do identification knowledge sources perform that are trained on multi-conditional data containing samples from different conditions and scenes?

4.2.2 Data Sets and Preprocessing Methods

The NIGENS database currently consists of 12 classes of everyday sounds (engine, crash, footsteps, piano, dog, phone, knock, fire, crying baby, alarm, female speech, and a general sound class). All sound files were manually annotated for onsets and offsets of target sound events.

Sound files were decomposed into overlapping blocks of 500 ms. As a basis for feature generation, we used the following auditory representations provided by the Auditory Front-end (AFE)[1]:

- ratemaps: auditory spectrograms that represent auditory nerve firing rates for each time frame (20ms) and individual gammatone frequency channel (computed by smoothing the corresponding inner hair cell signal representation with a leaky integrator),
- spectral features: 14 different statistics like flatness, kurtosis, etc., that summarise the spectral content of the ratemap for each time frame,
- onset strengths: measured in decibel for each time frame and frequency channel, calculated by the frame-based increase in energy of the ratemap representation,
- amplitude modulation spectrograms: each frequency channel of the inner hair cell representation is analysed by a bank of logarithmically-scaled modulation filters, so that for each time frame there are *number of frequency channels* $\times$ *number of modulation filters* values.

The AFE representations (16-channel ratemaps, spectral features (built over 32 channels), 16-channel onset strengths maps, and 8×9-channel amplitude modulation maps) were averaged over the left and right channel. Features were then calculated by applying the following operations:

- For each 500 ms block, we computed the L-statistics (L-mean, L-scale, L-skewness, L-kurtosis) of the representations over time.
- We also computed the first two deltas of the representation over time, corresponding to the discrete derivatives, and calculated the L-statistics on these as well.

1 `http://twoears.aipa.tu-berlin.de/doc/1.0/afe/`

This procedure gave rise to 154 spectral features, 176 ratemap features, 576 amplitude modulation spectrogram features, and 176 onset strength features. The resulting 1,082 elementary features (or proper subsets of them) then served as input for the classification procedures. For further details please refer to section 3.2.

4.2.3 Methods

Hard Classification

Method (1): lasso-fs1 Lasso is a classification method with an embedded feature selection procedure. It is based on a linear logistic regression model with an L1 penalty for the regression coefficients. This penalty forces many regression coefficients to be zero, leading to sparser models. An important factor determining the sparsity of the final model is the strength of the L1 regularization term, which is controlled by the regularization parameter. To adjust its value, we conducted 5-fold cross-validation on the training set for all 100-candidate values from the regularization path, and chose the value with the best cross-validation performance (cf. eq. (3.1)).

Method (2): svm-fs1 We also employed a two-stage procedure, where lasso was used to select two different sets of features (fs1 and fs3, cf. section 3.3.2) with non-zero coefficients at a particular λ, which was determined via cross-validation on the training set.These features were then handed as input features to a linear support vector machine (svm), which was used for classification (please refer to section 3.3 for details). The hyper-parameter C of the SVM was adjusted by using 4-fold cross-validation on the training set.

Probabilistic Classification

Method (3): Mixtures of factor analysers We developed a variant of the "generative" mixture of factor analysers approach for high-dimensional data which is particularly useful if the number of training data is small. Let $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$ denote a set of observations where each observation, $\boldsymbol{y}_n$, is a D-dimensional column vector with elements y_{dn}, $\forall\ d = 1, \ldots, D$, such that $\boldsymbol{y}_n = (y_{1n}, \ldots, y_{Dn})^\top$. We assume that each D–dimensional data vector $\boldsymbol{y}_n$ is generated by first linearly transforming a $K < D$ dimensional vector of unobserved independent Gaussian sources $\boldsymbol{s}_n = (s_{1n}, \ldots, s_{Kn})^\top$ followed by adding D–dimensional zero-mean Gaussian noise $\boldsymbol{e}_n = (e_{1n}, \ldots, e_{Dn})^\top$, independent between dimensions, and with the same precision r across all dimensions. This is mathematically

expressed as:

$$\boldsymbol{y}_n = \boldsymbol{U}\boldsymbol{s}_n + \boldsymbol{e}_n = \left(\sum_{k=1}^{K} \boldsymbol{u}_k s_{kn}\right) + \boldsymbol{e}_n, \tag{4.7}$$

where $\boldsymbol{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K)$ is a unit-norm normalized linear transformation matrix named the *normalized factor loading matrix*. Each element of the normalized factor loading matrix is a D-dimensional column vector, shown as $\boldsymbol{u}_k = (u_{1k}, \ldots, u_{Dk})^\top$, $\forall k = 1, \ldots, K$, satisfying $\|\boldsymbol{u}_k\| = 1$. We assume each observation is generated from a parametric mixture of Bingham–Gaussian factor analysers shown as:

$$p(\boldsymbol{Y} \mid \underline{\tau}, \underline{\boldsymbol{S}}, \underline{\boldsymbol{U}}, \underline{r}) = \prod_{n=1}^{N} \sum_{i=1}^{I} \tau_i \; p(\boldsymbol{y}_n \mid \boldsymbol{s}_{in}, \boldsymbol{U}_i, r_i), \tag{4.8}$$

where $\underline{\tau}$ denote a set of mixing proportions.

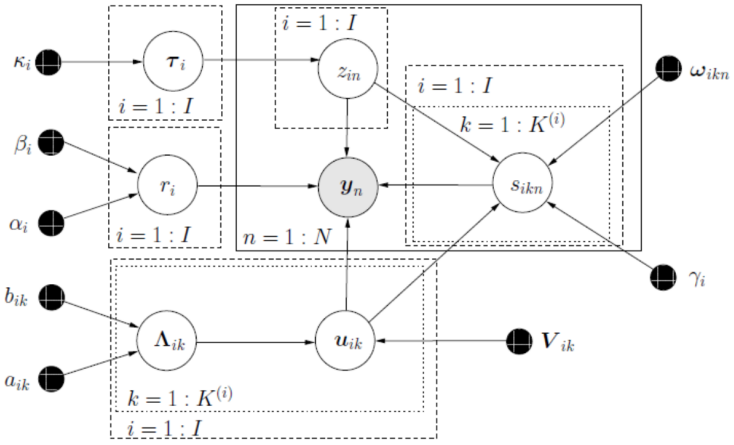


Figure 4.2: Directed acyclic graph representing the Bayesian mixtures of Bingham–Gaussian factor analysers. Random variables are denoted by open circles, and deterministic parameters (hyperpriors) are denoted by smaller solid circles. The shaded open circle denote the observed variables. Edges denote possible dependence, and plates denote replications.

Inference is done using Bayesian methods, because it allows better handling of model complexity compared to maximum-likelihood inference. The graphical model of Fig. 4.2 depicts the chosen dependencies. Our choice of conjugate prior distributions are: (1) A Gaussian-Dirac prior distributions over the conditional distribution of the source factors given the latent variables and the normalized factor loading matrix, (2) Bingham prior distributions over the columns of the normalized factor loading matrix, (3) a single Gamma distribution as a hyperprior distribution over the largest eigenvalue of each column of the factor loading matrix, (4) Gamma prior distribution over noise precision, and (5) sparse Dirichlet prior distributions over the mixture proportions. We showed that the chosen prior

distributions are conjugate to the data likelihood (Taghia, Trovitzsch, Obermayer (2015), Bayesian Bingham-Gaussian Factor Analysis, under review), as the result the posterior distributions after optimization will have the same functional form as their prior. We then used variational inference to optimize the posterior distributions.

4.2.4 Auditory Machine-Learning Training and Testing Pipeline

To facilitate the development and comparison of algorithms, the training of new models with different feature sets and different learning methods, we have implemented a highly flexible training pipeline[2] tailored to the needs of offline training for block-based sound source-type classification. This pipeline among others has the following features:

Multi-conditional auditory scene simulation. Ear signals are produced from audio files using the Binaural Simulator from the Two!Ears system. This can be done under various conditions (for instance with or without reverberations, with or without interfering sources). An arbitrary number of such conditions can be specified and will be simulated to produce multi-conditional training data. The motivation to do so instead of training on "clean" data is to include invariance to different conditions into the model by using data that enforces this.

Auditory Front-end (AFE). From the ear signals, the features are produced employing the AFE in exactly the same manner as in the Two!Ears system. This is to ensure that training and application of the models are done under the same conditions.

Automatic intermediate results saving. All products from intermediate stages, such as the ear signals or features produced by the AFE, are saved together with their configurations, since these intermediate stages can be very time-consuming (order of weeks, for the NIGENS database, for instance). They are saved in such a way that they will be recombined automatically whenever parts of a configuration have already been computed before. Training data can be exported for use outside of the pipeline.

Interface for feature creators. The whole pipeline is constructed in an object-oriented way, and with flexibility in mind. In particular, it is easily possible to create new feature sets that can be plugged into the pipeline by just implementing a class that inherits from a *feature creator* superclass. Those feature sets are constructed in a way that automatically delivers a detailed description of each feature dimension. Feature masks can additionally be used to incorporate results of feature selection methods into the training process.

Interface for model trainers. Similar to the case of the feature creators, there are interfaces

2 `https://github.com/TWOEARS/identification-training-pipeline`

for *model trainers*. Any class inheriting from them can implement its own technique (like lasso or svm), and can be plugged into the pipeline without modification. Wrapper trainer classes for conducting stratified cross-validation, e.g. for hyper-parameter search, are provided.

(Cross-)Evaluating models. The pipeline can be used tu evaluate trained models on auditory data. In particular, it is easy to create cross-testing configurations to test models under conditions different from training, in order to evaluate robustness.

Plug and play. Models created by the training pipeline implement a model interface, and can be plugged directly into the Two!Ears system – i.e. into the "knowledge sources", feeding the system with source-type hypotheses.

4.2.5 Classification Results: Hard Classification

In the first experiment we evaluated the identification knowledge sources on the data from task 2 (target sounds embedded in white noise, cf. section 4.2.1). We estimated the generalization performance of sound-type classifiers (target sound against the rest) for all 11 sound classes from the NIGENS databases, which were trained on data from the training set with a particular signal-to-noise-ratio (SNR) and tested on the test set data with the same SNR. This means that we trained specialized knowledge sources for the particular noise condition, and tested these knowledge sources on the condition of their expertise. Figure 4.3 shows the performance for lasso (lasso-fs1, shown in red), and the two-step learning scheme (svm-fs1, shown in blue).

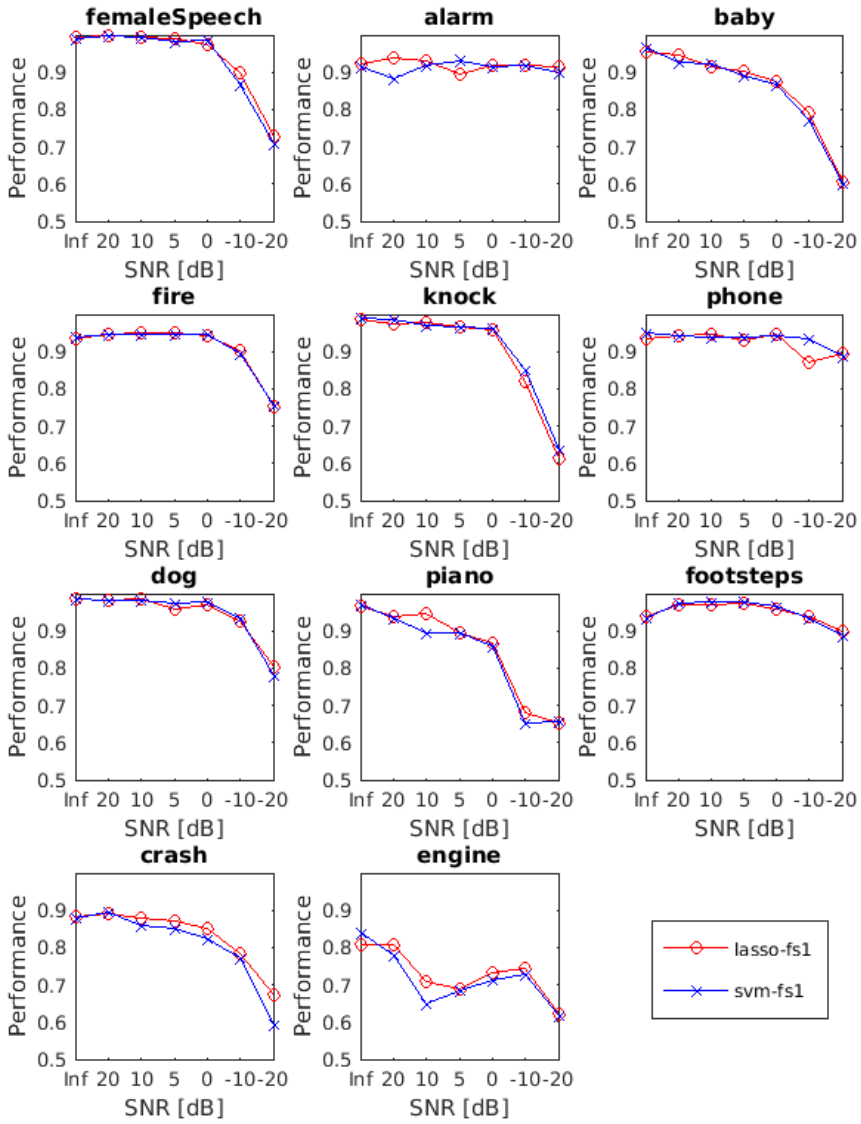


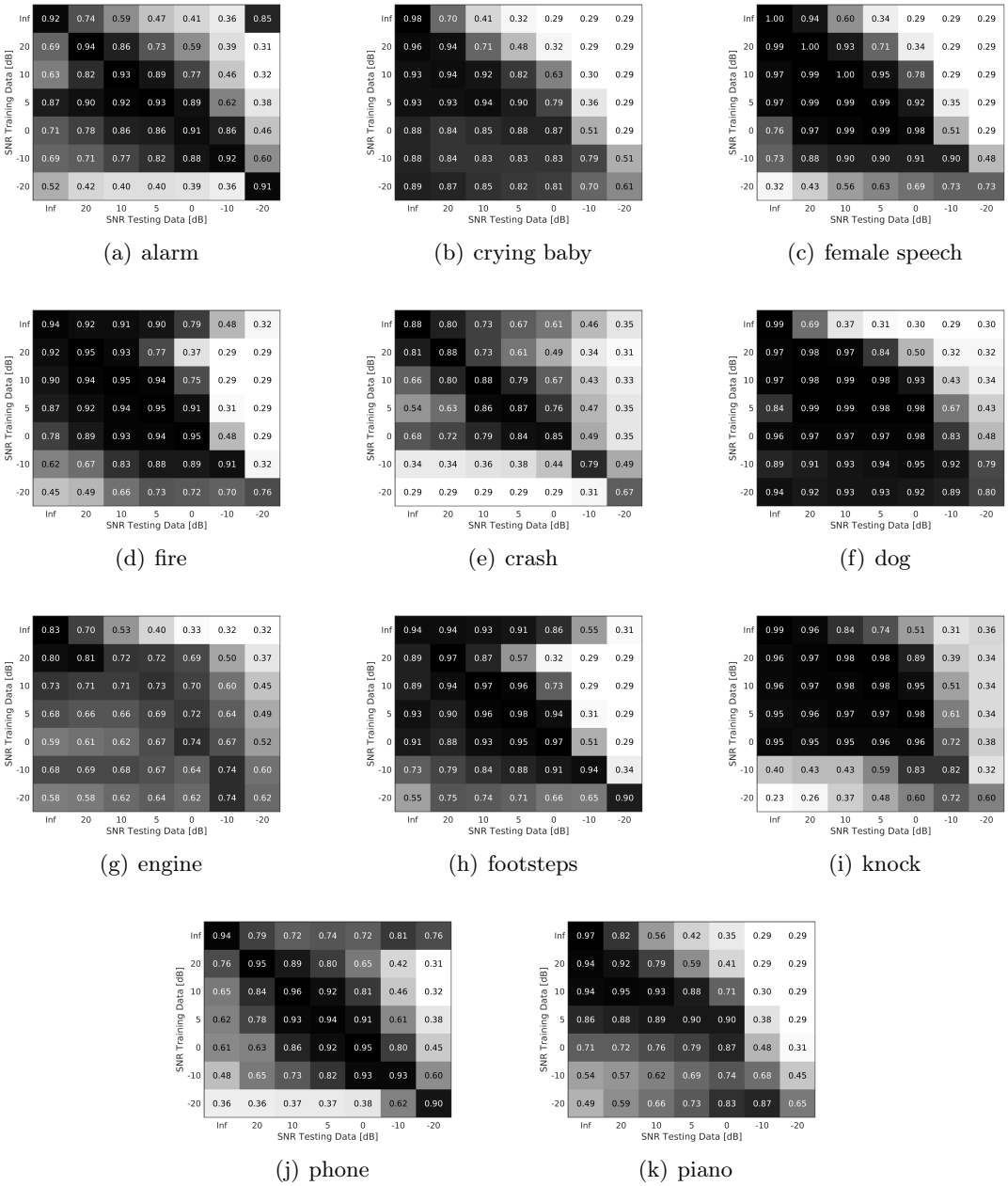
Figure 4.3: Performance of sound-type classifiers constructed with lasso-fs1 and the two-step learning scheme svm-fs1 using the monaural feature set. Classification was performed on features computed on frames of 500ms of auditory scenes, where "dry" sounds played from an azimuth of 0^o were superimposed with ambient white noise (task 2, single-condition training, cf. section 3.3.5). Performance values are shown for the iso-testing condition as a function of the different SNR values.

For high SNRs (no noise - 20 dB), classification performance ranges from 1.0 for female speech to 0.78-0.85% for the most difficult-to-classify sound-type "engine". Statistical testing of pairwise performance differences over all noise levels and classes showed that lasso-fs1 performed significantly better that svm-fs1 (one-sided t-test, $\alpha = 0.05$, $p = 0.026$). However, the size of this performance improvement is very small, as can be seen from the alignment of the red and blue curves in Fig. 4.3, which is, in most cases, very close.

One can roughly distinguish three categories of sound types based on the performance profiles in Fig. 4.3. For the first group, the performance of the specialized knowledge sources is still quite good even at very high noise levels. This group includes the sound types "alarm", "phone", and "footsteps", where the knowledge sources could be trained for all noise levels (up to the most extreme value of -20 dB) to give performances around 0.9, or better. For the second group, which comprises "fire", "dog", "female speech", and "knock", the classifiers seem to be able to cope well with values up to 0dB, where the level of the noise is equal to the level of the signal. However, for higher noise levels (-10 dB and -20 dB) the classification performance quickly deteriorates. The third group consists of the sound types "crying baby", "piano", "crash", and "engine". For these sound types, classification performance decreases more gradually with decreasing SNR. These results were all obtained by specialized knowledge sources for a particular sound type and noise level.

We were then interested how well the individual knowledge sources classified data from a different noise-level then they had expertise in. In a second experiment we therefore analysed the cross-testing performance of the classifiers trained for the first experiment across different noise levels. The results for lasso-fs1 are shown in Fig. 4.4, the results for svm-fs1 in Fig. 4.5. Each panel shows the iso- and cross-testing performances for one of the eleven sound types, where each matrix entry contains the performance of model that was trained at a particular SNR (rows) and tested at another SNR (columns). The diagonal entries therefore correspond to the iso-testing conditions, the off-diagonal element to the cross-testing conditions.

In contrast to the iso-testing case, there was no significant difference in performance between lasso-fs1 and svm-fs1 (one sided t-test, $\alpha = 0.05$) for the cross-testing conditions across all classes and noise level combinations. Knowledge sources performed increasingly worse on noise levels that were higher than the noise they were trained on. If the experts were tested on lower noise levels than the one they were trained on, this decrease in performance was in general less pronounced. For instance, for the case of the sound type "dog", a classifier trained at the most extreme noise level (SNR -20 dB) still showed an excellent performance at the other extreme of zero noise. However a classifier trained for the same sound type at zero noise, dropped in performance to 0.42 at a still relatively high SNR of 5 dB.



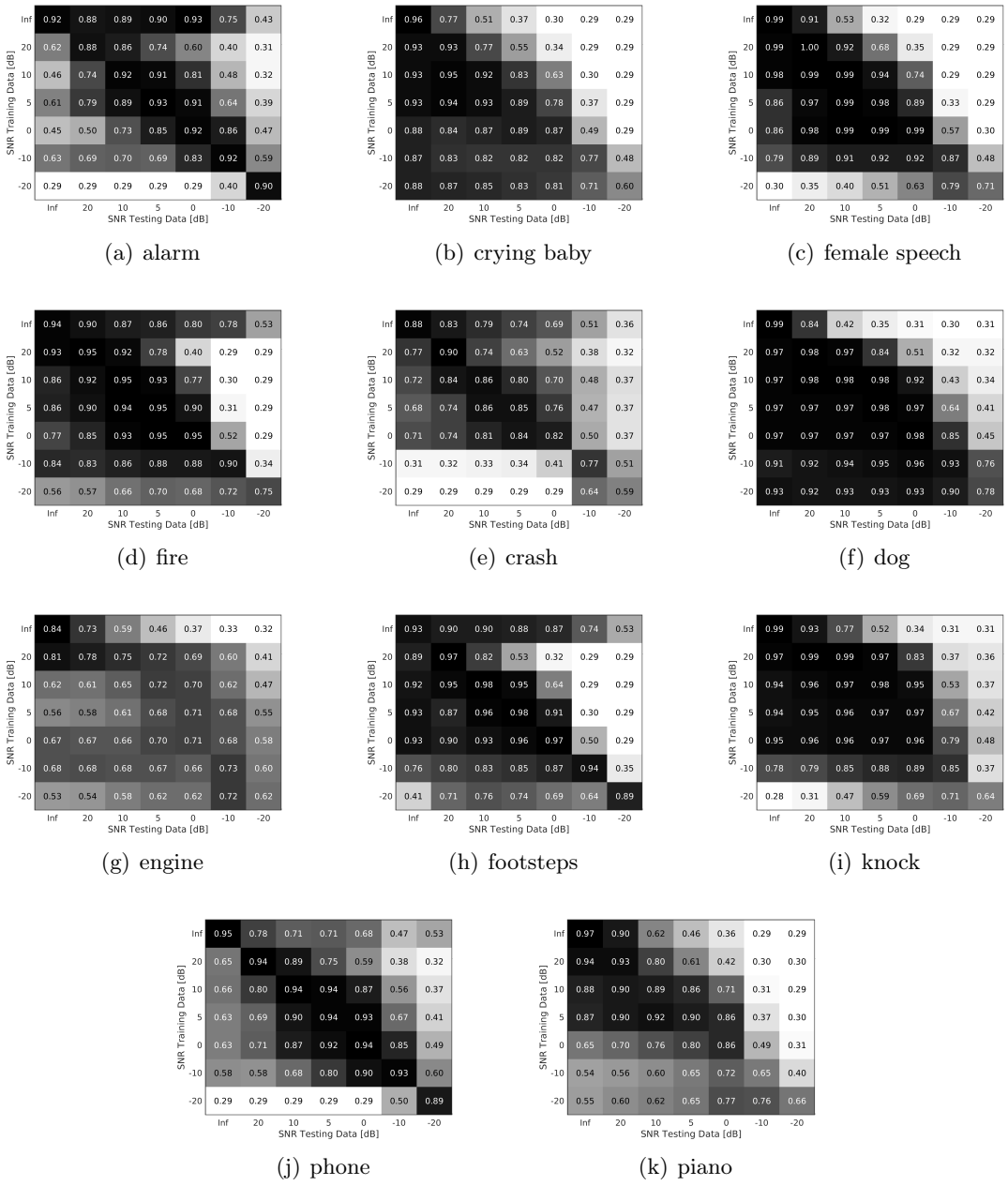
(a) alarm (b) crying baby (c) female speech

(d) fire (e) crash (f) dog

(g) engine (h) footsteps (i) knock

(j) phone (k) piano

Figure 4.4: Iso- and cross-testing results of classifiers trained with the lasso learning scheme (criterion fs1) on the monaural feature set for every SNR and for all 11 sound classes. Vertical and horizontal axes correspond to the SNRs used for training and test. Matrix entries denote the corresponding prediction performances (cf. eq. (3.1)). In addition, values are visualized by a grey level encoding from black (good performance) to white (bad performance).



(a) alarm (b) crying baby (c) female speech

(d) fire (e) crash (f) dog

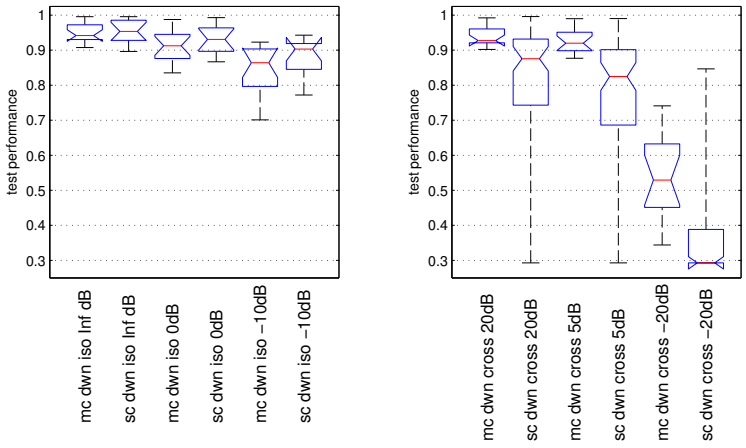
(g) engine (h) footsteps (i) knock

(j) phone (k) piano

Figure 4.5: Same as Fig. 4.4 but for classifiers that were trained under the machine learning scheme svm-fs1.

Quite the opposite happens for “crash”, “phone”, “knock”, “alarm”, and “female speech”, where classifiers trained at the highest noise level (SNR −20 dB) do already perform much worse at slightly lower noise levels. For some sound types, on the other hand, training at zero noise seems to generalize reasonably well to higher noise levels. Examples are the classifiers for “footsteps”, “ fire” for both classification methods, “phone” for lasso-fs1, as well as “alarm” and “crash” for svm-fs1.

In a third experiment, we analysed the impact of multi-conditional training on the classification performance of the knowledge sources for different levels of added white noise. Classifiers were trained on data from a subset of conditions of all three sound tasks that were described in section 4.2.1. Here, we tested the resulting classifiers on data from task 2, which was also used in the previous experiments.



(a) iso (b) cross

Figure 4.6: Performance of classifiers constructed using multi-conditional (mc) training in comparison with the performance of classifiers constructed using single-conditional (sc) training. Specialised classifiers are trained on data from task 2 (added ambient white noise at different SNRs). (a) In the iso-testing condition, the generalization performance was evaluated at three different SNRs (no noise, 0 dB, and −10 dB) that were among the conditions included in the multi-conditional training. (b) In the cross-testing condition, performance was evaluated at different (20 dB, 5 dB, and −20 dB) SNR values that were not part of the multi-conditional training set. Boxes summarise data across the two classifiers lasso-fs1 and svm-fs1, across three different base feature sets, and across the four different classes “alarm”, “crying baby”, “female speech”, and “fire”.

Figure 4.6 summarizes the performance of classifiers constructed using single-conditional in comparison to multi-conditional training. The specialised knowledge sources show a superior performance when evaluated for data from the same condition, but performance drops below the values of multi-conditional trained knowledge sources for the cross-testing condition.

Knowledge sources constructed by multi-conditional training thus show a better generalisation performance across testing conditions and should be preferred for the Two!Ears system under general conditions. If, however, information about conditions is available to the system, specialised classifiers may be selected, for example via feedback mechanisms, leading to an overall improvement in sound-type recognition.

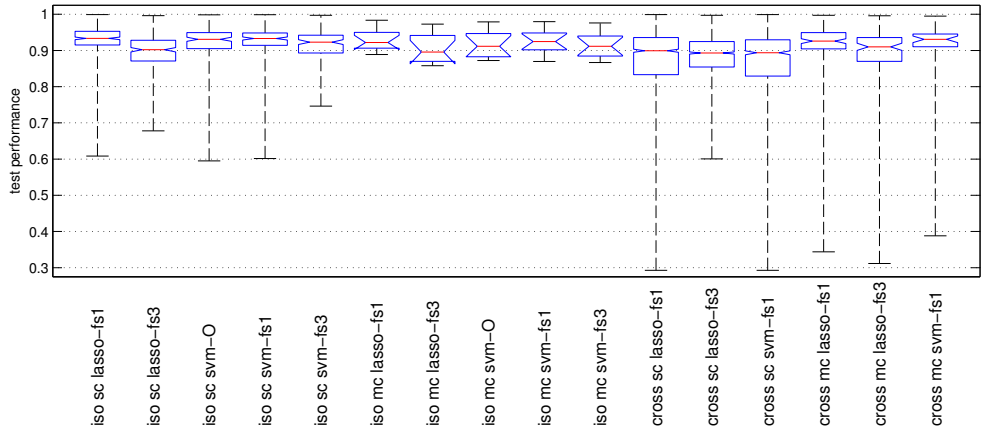
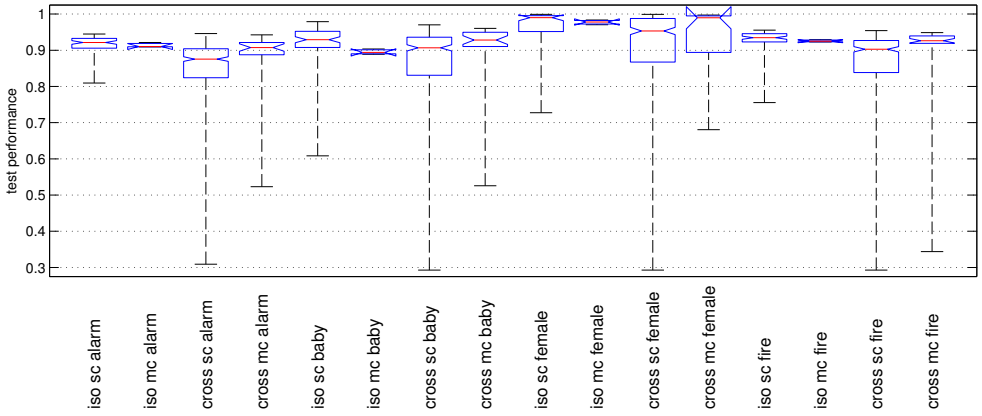
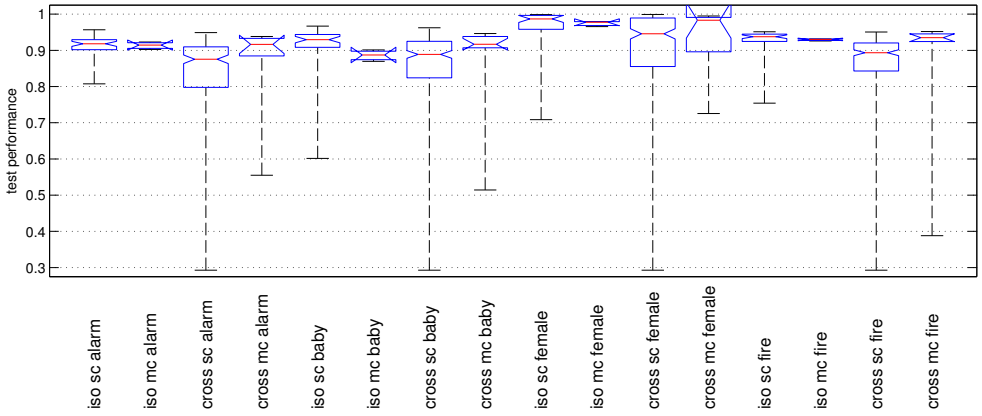


Figure 4.7: Performance of classifiers constructed using single-conditional ("sc") in comparison with multi-conditional ("mc") training for the iso- and cross-testing conditions. The figure shows the performance values for the five classification schemes lasso-fs1, lasso-fs3, svm-O (no feature selection), svm-fs1, and svm-fs3 for the iso-testing condition. The cross-testing conditions do not include svm-O and svm-fs3. Each box plot summarises results across the four classes "alarm", "crying baby", "female speech", and "fire", across all base feature sets, and across all tasks and conditions.

Fig. 4.7 and Fig. 4.8 show the performance of classifiers constructed using single-conditional in comparison with multi-conditional training for the iso- and cross-testing conditions. Comparing the boxes of Fig. 4.7 we see, that the classification performance is similar across the different learning schemes with the exception that there is a tendency for feature selection according to criterion fs1 to provide slightly better results. If we, however, compare the iso- with the cross-testing conditions there is a significant loss of performance for the classifiers which underwent single-condition training. For classifiers constructed using multi-conditional training the loss of performance is not significant, i.e. they show a better generalisation to unseen conditions. As a result, the performance of classifiers trained with multi-conditional data is significantly better than the performance of classifiers trained with single-conditional data for the cross-testing condition. Figure 4.8 confirms above conclusions separately for the four classes "alarm", "crying baby", "female speech", and "fire".



(a) lasso-fs1



(b) svm-fs1

Figure 4.8: Performance of classifiers using single-conditional ("sc") in comparison with multi-conditional ("mc") training for the iso- and cross-testing conditions. *(a)* Summary of performance values for the lasso classification scheme (criterion fs1) shown separately for the four different classes. Each box plot summarises results across all base all feature sets, all tasks and conditions. *(b)* Same as (a) for the two-stage classification scheme svm-fs1.

4.2.6 Classification Results: Mixtures of Factor Analysers

We built frame-based classifiers for sounds taken from the four classes "crying baby", knock", "fire", "piano" using sounds from all other categories as negative examples.

We assembled a high-dimensional set of features extracted from sound streams using the *Auditory Front-end* in the Two!Ears system. The feature set was constructed from the following cues, which were calculated over small windows of 20ms with a shift of 10ms:

- Frequency domain low-level statistics (centroid, crest, spread, entropy, brightness, high-frequency content, decrease, flatness, flux, kurtosis, skewness, irregularity, rolloff, variation),
- ratemap magnitudes of 32 frequency channels,
- amplitude modulation maps, and
- onset strength maps.

Additionally, the first (discrete) derivative of these cues over time was used. Feature vectors were then calculated by extracting L-statistics (L-mean, L-scale, L-skewness, L-kurtosis) of the above listed cues over blocks with a length of 500ms and a frame-shift of 167ms. The final feature set had a dimensionality of 846. The twelve categories comprised 837 sound files, which after feature extraction made for a set of approximately 10^6 samples. Probabilistic model outputs (target sound present vs. target sound absent) were transformed into class labels depending on whether the predictive probability was larger than 0.5, and models were evaluated by calculating the balanced accuracy on a hold-out (test) set.

Figure 4.9 shows the performance of the modified mixture of factor analysers (MBGFA) in comparison with the standard method (VB-MFA). Classification accuracy of MBGFA becomes superior to the one by VB-MFA if less training data are available. The extra gain in performance can be explained by the fact that MBGFA makes fewer factorizations during Bayesian inference compared to the full factorized approach of VB-MFA.

4.2.7 Conclusions

We have shown, that machine learning techniques allow us to construct classifiers for sound type with a very good overall classification performance, given that classification was performed on short blocks without using any contextual information. As expected, classifi-

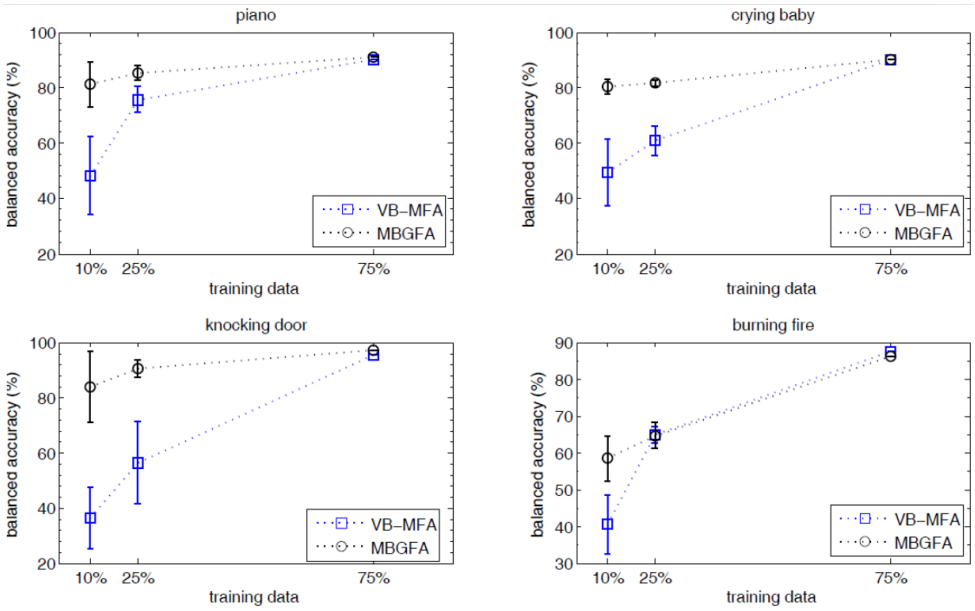


Figure 4.9: Classification accuracy and standard deviation, in terms of the average balanced accuracy, for various train/test splits of MBGFA and VB-MFA algorithms and for the four different sound categories. The total amount of training data is about 10^6. For the case with 10% training data, 10 splits are considered, and for the cases with 25% and 75% of training data, we have considered respectively 4 and 1 splits.

cation performance degrades with the strength of noise and of any distractor signal which is simultaneously present in an auditory scene. Still, classification performance remains high on average down to a signal-to-noise ratio of approximately 0 dB.

We then compared the performance of specialized knowledge sources with the performance of classifiers which were trained on more diverse training sets. We found that multi-conditional training was suitable to obtain classifiers that are robust over a wide range of different conditions, at the cost of only a slight decrease in performance compared to the specialized single-condition classifiers.

The probabilistic learning scheme using mixtures of factor analysers was extended to be also applicable to smaller training set sizes. It may be a useful scheme for constructing event experts for sound-type classification which also provide confidence values. However, training times are much longer and the number of data necessary for training is still much higher when compared to the hard classification methods. Therefore, an extension of svm-based classifiers to provide confidence values will be more practical.

4.3 Labeling objects based on vision

Vision can be used as a knowledge source in order to detect, identify, track and localize objects or humans. The work conducted on visual functions has been twofold, with the aim to integrate them in the deployment system on the basis of a stereovision sensor mounted on the *KEMAR* head-and-torso-simulator (HATS) in an anthropomorphic configuration.

On the one hand, we tested several methods of the literature so as to select a robust and efficient subset suited to Two!Ears. The results of this evaluation of open-source off-the-shelf software is summarized in Deliverable 5.2. First, it led to the selection of a strategy for the monocular detection and tracking of multiple humans standing upright. By incorporating the calibrated model of the stereoscopic sensor, a coarse 3D information on humans in the scene can be recovered. Deliverable 5.2 also presents results on object perception on the basis of a 3D point cloud. To begin with, this 3D data was provided by RGB-D device, typically the *Xtion Pro Live* sensor with a resolution of 640x480 pixels, in the principle similar to the *Microsoft Kinect*. When the stereoscopic sensor is fully working, a 3D point cloud will be extracted from its images. Using several open source packages (*ORK*, *TableTop*, *Linemod*,...), a set of objects could be learnt from an image database acquired offline, typically from several hundreds of images acquired from different viewpoints around them. Then, the online identification and location of one of these objects inside a query image could be performed using some retrieval method. For instance, assuming that these objects lie on a table, a classical segmentation algorithm (*e.g.*, the one included in the *TableTop* package) allows their separation from the supporting plane. To a larger extent, though not envisaged so far, humans could be detected and tracked (in terms of location and posture, though the attitude is not required) by using the open-source *OpenNI/NiTE* libraries.

On the other hand, two more fundamental specific contributions were proposed. The first one (Manfredi *et al.* (2015a)) aimed at simplifying the models of textured objects (thus using also RGB information). Several processes were compared to build their learning databases. An optimal sampling method was proposed to minimize the numbers of required images, while preserving the recognition and localization performances.

The second contribution concerned the localization of structureless rigid objects (Manfredi *et al.* (2015b)). A single RGB image was used during the offline learning step (Figure 4.10(a)), so that this image could be taken from the Internet for example. Then, online, a depth image and the matching of some interest points were used in order to generate the RGB+depth information on the object (Figure 4.10(b)). A 3D localization was then proposed, as accurate as the conventional PnP procedure of the litterature (Figure 4.10(c)(d)).

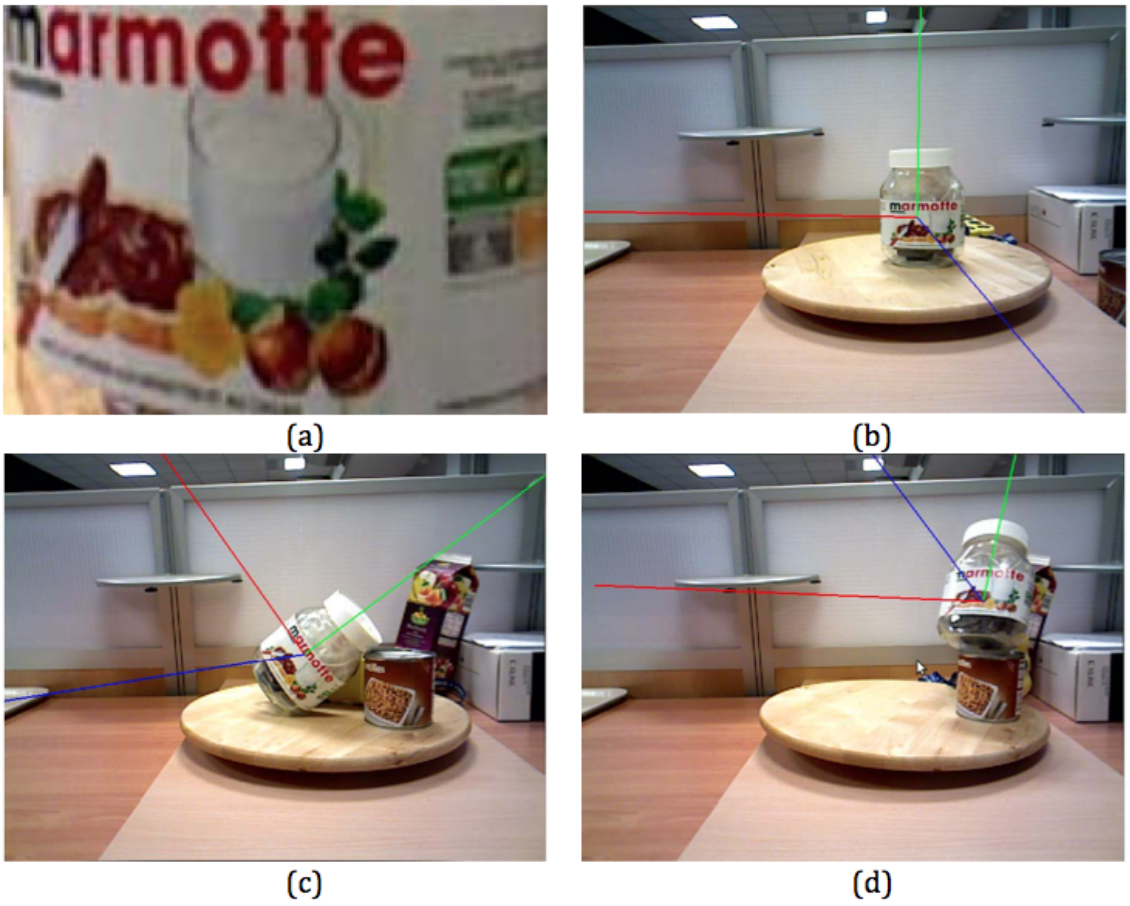


Figure 4.10: Object modeling and localization. (a): Offline RGB image. (b): Online 4D model. (c)-(d): Online object localization.

4.4 Audio-visual speech recognition

Audio-visual integration was used to more reliably recognize keywords. This is of interest in variable and acoustically challenging conditions, such as those considered in the CASA scenario. For this purpose, we employed the CHiME challenge data, which is realistic, binaural acoustic data recorded in a home, including environmental noises such as background music, household appliances, children playing, etc. The CHiME data contains signal-to-noise ratios between -6 dB and 9 dB Barker *et al.* (2013) and it was used in combination with matching video data from the GRiD corpus Cooke *et al.* (2006).

Two different approaches were considered; direct concatenation of audio and video features, and joint recognition by using an appropriate graphical model.

4.4.1 Feature concatenation

In the first experiments we assume the acoustic model and the video model are always synchronous. A straightforward approach to integrating audio-visual information for automatic speech recognition is direct concatenation of audio and video features. The concatenated feature vectors can then be jointly learned by using suitable models such as Gaussian mixture models or a deep neural network.

We used the Kaldi toolkit (`http://kaldi-asr.org`) for speech recognition with concatenated audio-visual features. For mel-frequency cepstral coefficient (MFCC) audio features were concatenated with corresponding video features frame-by-frame before computing the delta and acceleration features. The concatenated features were then used to train triphone acoustic models (GMMs) with linear discriminant analysis (LDA) and maximum likelihood linear transform (MLLT) feature transforms. Speaker-independent models were created first by training models on all speakers before training speaker-dependent models with speaker adaptive training.

Table 4.2: Keyword recognition accuracies (%) of various systems for the *development* test set. Models were trained on clean data.

	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
Video	63.27	63.27	63.27	63.27	63.27	63.27	63.27
Audio	47.79	47.79	60.71	71.26	81.04	85.03	65.60
AV Concat.	68.62	72.19	75.43	81.38	84.18	85.88	77.95

Table 4.3: Keyword recognition accuracies (%) of various systems for the *development* test set. Models were trained on mixed data.

	-6 dB	-3 dB	0 dB	3 dB	6 dB	9 dB	Average
Video	63.27	63.27	63.27	63.27	63.27	63.27	63.27
Audio	61.99	66.92	76.70	83.84	87.33	91.58	78.06
AV Concat.	77.38	79.68	81.21	82.65	84.52	85.37	81.80

Tables 4.2 and 4.3 list keyword recognition accuracies (letter and digit) of the feature concatenation system on the CHiME Challenge data Barker *et al.* (2013). For comparison, each table also includes recognition accuracies of systems employing audio or video features only. For Table 4.2, the systems were trained on noise-free ('clean') data. While the video-only system achieves a keyword accuracy of 63% regardless the SNR, the audio-only system performed significantly worse at SNRs lower than 3 dB and substantially better at SNRs above 3 dB. The video features and the audio features are complimentary in this case, as the concatenated system outperformed individual systems consistently across all SNRs. This is especially the case for low SNRs, e.g. at the 0 dB SNR, the audio-only recognition

accuracy is 61% and the video-only accuracy is 63%, but the combined performance is 75%.

It has been widely shown that by training the acoustic models on noisy data, i.e. the multi-conditional training (MCT), the speech recognition accuracy can be significantly improved. Table 4.3 lists results using multi-conditional training which included noisy data at a range of SNRs between -6 dB and 9 dB for training the acoustic models. It can be clearly seen that MCT substantially improved the audio-only performance, with on average a 13% absolute accuracy improvement. However, the benefit of audio-visual integration became only apparent for SNRs below 3 dB. For SNRs higher than 3 dB, the accuracies of the feature concatenation system are worse than those produced by the audio-only system.

One possible explanation is that simple concatenation of audio and video features applies equal weights to both audio and video streams. At higher SNRs where the audio-only performance is significantly better than the video-only performance, applying equal weights to both audio and video streams cannot be optimal. A better alternative would be to apply adaptive weights depending on the SNRs, so that at higher SNRs the audio stream would contribute more than the video stream.

4.4.2 Feature fusion using state-space modeling

Whereas feature concatenation is only applicable if it can be assumed that the acoustic model and the video model are always synchronous, other inference methods can also be employed for loosely synchronized audiovisual data. For speech data specifically, it has been shown that, due to preparatory movements in the speech production process, the visual modality information can precede that of the acoustic modality by up to 120 ms Luettin *et al.* (2001).

Therefore, in the second set of experiments, we assumed only synchrony within phonetic units, and evaluated this idea by decoding via token passing within the coupled HMM framework (for real-time performance) and by turbo decoding (by multiple iterations) Scheler *et al.* (2012), Zeiler *et al.* (2016).

In addition to considering a range of different decoders, experiments were also done using two different types of so-called observation uncertainties, where the feature vectors are not taken to correspond to ground-truth, but are rather considered as point-estimates of random variables, cf. e.g. Kolossa and Haeb-Umbach (2011). Two types of observation uncertainty handling were considered:

- uncertainty decoding, where feature uncertainties are added to model variances, as derived from a marginalization over the hidden clean speech Deng *et al.* (2005) and

- noise-adaptive LDA (NALDA), where at each time-step an optimal linear transform of the current feature vector is derived by maximizing class discriminance Kolossa *et al.* (2013).

The results of these experiments are shown in Table 4.4.

SNR	-6	-3	0	3	6	9	avg.
Video	69.98	69.98	69.98	69.98	69.98	69.98	69.98
Audio NoUnc	71.90	79.05	82.56	87.75	91.64	91.60	84.08
Audio UD	72.99	77.57	81.60	88.73	91.49	91.74	84.02
Audio NALDA	74.00	78.94	85.19	90.93	92.40	93.30	85.79
CHMM NoUnc	84.72	85.81	88.68	90.47	91.22	92.09	88.83
CHMM UD	83.63	84.59	87.77	88.97	91.18	90.64	87.80
CHMM NALDA	84.13	87.59	90.28	92.40	93.36	93.43	90.20
Turbo Diag	85.75	88.58	90.45	92.16	93.68	93.52	90.69
Turbo UD	84.34	87.57	89.67	91.48	93.60	92.71	89.89
Turbo NALDA	**87.21**	**89.48**	**92.08**	**93.09**	**95.26**	**95.12**	**92.04**

Table 4.4: Results with estimated uncertainties. Best results are marked in bold.

As can be seen, the iterative approach of turbo decoding performs best. In almost all cases, considering observation uncertainties by NALDA contributes markedly to the overall performance, and audio-visual recognition always outperforms the better of the single modalities.

4.5 Graphical modelling approaches

Several approaches for including graphical models (GMs) into the Two!Ears framework have been investigated in the recent project period. GMs are especially appealing in the context of computational auditory scene analysis (CASA), because they provide a probabilistic representation of variables and labels of interest. Hence, the general approach in Two!Ears is to integrate GM-based techniques into all relevant system components. So far, several approaches utilising GMs have been considered, including

- the application of recursive Bayesian estimation techniques described in Sec. 4.1.2,
- the use of GMs for source segmentation as described in Sec. 5.2.2,
- approaches to refine segmentation results by introducing GM based techniques from the field of computer vision,
- initial ideas for utilising GMs for high-level CASA using semantic labels.

As the first two approaches are being described in the corresponding sections, only the latter two will be covered in more detail here.

4.5.1 Source segmentation based on Markov random fields

The source segmentation stage is an essential building block of the TWO!EARS system architecture. It aims at assigning subsets of auditory features to individual sound sources. This will allow to feed higher-level processing components (like classifiers for sound type recognition) with separated, source specific audio streams. In Sec. 5.2.2 of the following chapter, a model for this task is introduced which performs clustering on spatial cues by assuming that observed source positions were generated from underlying circular probability distributions. However, the proposed approach performs segmentation individually on each time-frequency unit. Hence, an idea to further refine this initial segmentation is to take local relationships between neighbouring time-frequency units into account. This approach is widely used for image segmentation tasks in the field of computer vision, see e.g. Ren *et al.* (2011) and Besbes *et al.* (2011) and has also been applied for the segmentation of audio signals, see e.g. Lagrange *et al.* (2007). A brief introduction into the general approach used in image segmentation via Markov random fields (MRFs) is given below.

Markov random fields in image segmentation. A common simplification for most state-of-the-art image segmentation techniques lies in the assumption that an image can be modelled as a set of interconnected random variables (RVs) that correspond to each pixel. A label is assigned to any pixel in order to symbolize the class that this pixel belongs to. For an image of size $N \times M$ pixels, each pixel is associated with a feature vector $\boldsymbol{x}_i,\ i = 1, \ldots, L$, where $L = NM$ is the total number of pixels in the image. All feature vectors can be grouped together in a feature matrix

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_L \end{bmatrix}.$$

Generally, features are assumed to be scalar values for grayscale images, and 3-dimensional RGB values for colour images. Note that arbitrary feature representations can be used in this framework. Typically, the number of labels or objects K in an image is assumed to be known a priori. Therefore, each label $\lambda_i,\ i = 1, \ldots, L$ is an integer value from the set of labels $\mathcal{L} = \{1, \ldots, K\}$. A possible labelling for a specific image can thus be represented as a vector

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \cdots & \lambda_L \end{bmatrix}^T.$$

Probabilistic image segmentation techniques aim at finding the optimal labeling as the maximum *a posterioi* (MAP) estimate for the labels

$$\hat{\boldsymbol{\lambda}}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\lambda}} p(\boldsymbol{\lambda} \mid \boldsymbol{X}), \tag{4.9}$$

given the observed features $\boldsymbol{X}$. By applying Bayes's rule, the conditional probability in optimization problem (4.9) can be rewritten as

$$\begin{aligned} p(\boldsymbol{\lambda} \mid \boldsymbol{X}) &= \frac{p(\boldsymbol{X} \mid \boldsymbol{\lambda})\, p(\boldsymbol{\lambda})}{p(\boldsymbol{X})} \\ &\propto p(\boldsymbol{X} \mid \boldsymbol{\lambda})\, p(\boldsymbol{\lambda}) \\ &= \prod_{i=1}^{L} p(\boldsymbol{x}_i \mid \lambda_i) p(\lambda_i) \\ &= \sum_{i=1}^{L} \log\Big(p(\boldsymbol{x}_i \mid \lambda_i)\Big) + \log\Big(p(\lambda_i)\Big) \end{aligned} \tag{4.10}$$

An exhaustive search to find the optimal labelling is computationally intractable, as the search space encompasses K^L possible labellings. Hence, solving Eq. (4.9) has to be tackled in a more sophisticated manner. Therefore, it is of particular interest to find appropriate representations for the log-likelihoods introduced in Eq. (4.10). This can be achieved by considering a MRF representation of the underlying image model.

The optimization problem in Eq. 4.10 is unconstrained, given the generally intractable search space that was described previously. Nevertheless, it is possible to impose certain constraints on the image segmentation task, which significantly reduce the dimensionality of the search space.
The fundamental assumption for these constraints is that the colour of a specific pixel does not depend on the colours of all other pixels in the image. Instead it is assumed that the pixel's colour is governed by local dependencies induced through neighbouring pixels only.

Narrowing down the structure of the underlying model to a set of local neighbourhoods surrounding individual pixels, the whole image can be represented as a MRF if the conditions

1. $\forall\, \lambda_i \;:\; p(\lambda_i) > 0$
2. $p(\lambda_i \mid \lambda_j,\, i \neq j) = p(\lambda_i \mid \lambda_j,\, j \in N_i)$

hold, where N_i denotes the set of neighbours of pixel i. The first condition states that label probabilities must always be greater than zero, because otherwise the joint probability of the MRF would be zero. The second condition specifies the local neighbourhood structure

in a way that allows to express the conditional probability of adjacent labels for pixels i and j through a neighbourhood variable N_i. Furthermore, a third condition is required: the underlying probability distribution $p(\boldsymbol{\lambda})$ has to fulfill the Hammersley-Clifford Theorem (Hammersley and Clifford (1971)). This theorem states, that a random field is a MRF, if and only if it obeys a Gibbs distribution

$$p(\boldsymbol{\lambda} \,|\, \boldsymbol{X}) = \frac{1}{Z} \exp\Big(- U(\boldsymbol{\lambda},\, \boldsymbol{X})\Big), \tag{4.11}$$

where $Z = \sum_{\forall \boldsymbol{\lambda}} \exp\Big(- U(\boldsymbol{\lambda},\, \boldsymbol{X})\Big)$ is a normalization constant. A clique is a subset of nodes in the graph, where each pair of nodes within this subset is neighbouring. A clique containing n nodes is called an n-th order clique, denoted by c_n. Image segmentation techniques usually restrict the image models to cliques of first and second order, called singletons and doubletons. It can be shown that considering singleton and doubleton clique potentials within a MRF, leads to an energy function

$$U(\boldsymbol{\lambda},\, \boldsymbol{X}) = \sum_{i=1}^{L} \Big(V_{c_1}(\lambda_i,\, \boldsymbol{x}_i) + \sum_{j \in \mathcal{C}_{\lambda_i}} V_{c_2}(\lambda_i,\, \lambda_j) \Big) \tag{4.12}$$

that is equivalent to the maximum likelihood expression of the optimization problem (4.10), see e.g. Ren *et al.* (2011). Having defined the energy function of the graph in terms of clique potentials as in Eq. (4.12), specific expressions of singleton and doubleton potentials can be derived.

Singleton clique potentials neglect any specific dependency between neighboring nodes in the graph. Thus, the energy associated with these cliques can be expressed as a likelihood that represents the achieved degree of data association. A straight-forward approach to model singletons is to assume Gaussian distributed features

$$p(\boldsymbol{x}_i \,|\, \lambda_i, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}_j|}} \exp\Big(- \frac{1}{2} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)\Big), \tag{4.13}$$

where D is the dimension of the feature space and $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ are the mean vector and covariance matrix associated with label $j = \lambda_i$, respectively. To simplify the notation, the label-specific parameters can be represented as a parameter set

$$\boldsymbol{\theta} = \{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}, \quad j = 1, \ldots, K.$$

Hence, singleton clique potentials can be expressed as the log-likelihood of the multivariate

Gaussian density (4.13), according to

$$\begin{aligned} V_{c_1}(\lambda_i,\, \boldsymbol{x}_i,\, \boldsymbol{\theta}) &\propto p(\boldsymbol{x}_i \,|\, \lambda_i,\, \boldsymbol{\theta}) \\ &= -\frac{D}{2}\log(2\pi) - \frac{1}{2}\log(|\boldsymbol{\Sigma}_j|) - \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_j). \end{aligned} \tag{4.14}$$

Additionally, the doubleton clique potentials serve as a smoothness-prior by favoring similar labels at neighbouring pixels. A simple approach used in many basic image segmentation algorithms is

$$V_{c_2}(\lambda_i,\, \lambda_j) = \beta\delta(\lambda_i,\, \lambda_j) = \begin{cases} -\beta, & \text{if} \quad \lambda_i = \lambda_j \\ \beta, & \text{otherwise} \end{cases}. \tag{4.15}$$

The parameter β is a scaling factor and determines the amount of smoothness that should be obtained during the image segmentation process. By imposing a large penalty on dissimilar labels within the doubleton cliques, the final segmentation tends to display smooth edges, and to include large homogeneous areas of similar labels. Setting the scaling parameter to a small value instead, the segmentation will yield sharper edges at the cost of being more sensitive to image noise.

By inserting Eqs. (4.14) and (4.15) into Eq. (4.12), the final energy function describing the image segmentation model can be derived as

$$U(\boldsymbol{\lambda},\, \boldsymbol{X},\, \boldsymbol{\theta}) = \sum_{i=1}^{L} \Big(-\frac{1}{2}\log(|\boldsymbol{\Sigma}_j|) - \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_j) + \sum_{j \in \mathcal{C}_{\lambda_i}} \beta\delta(\lambda_i,\, \lambda_j) \Big). \tag{4.16}$$

Hence, the image segmentation problem (4.9) originally expressed as a MAP optimization, can be re-formulated in terms of an energy minimization task

$$\hat{\boldsymbol{\lambda}}_{\text{MAP}} = \arg\max_{\boldsymbol{\lambda}} p(\boldsymbol{\lambda} \,|\, \boldsymbol{X},\, \boldsymbol{\theta}) = \arg\min_{\boldsymbol{\lambda}} U(\boldsymbol{\lambda},\, \boldsymbol{X},\, \boldsymbol{\theta}). \tag{4.17}$$

Solving Eq.(4.17) is still a challenging problem, but several sophisticated algorithms for approximate inference in MRFs are available, see e.g. Bishop (2006).

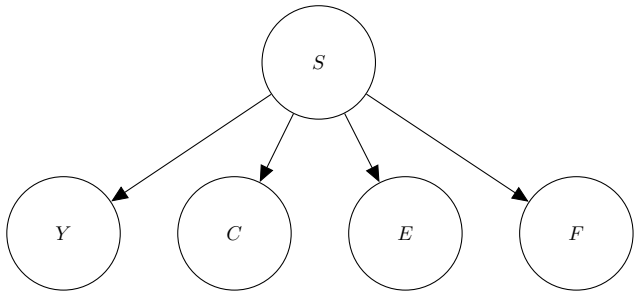
Application in TWO!EARS. Image segmentation and sound source separation can be naturally considered as very similar tasks. Segmenting a spectrogram into groups of time-frequency units belonging to different sources is essentially the same as assigning pixels in an image to dedicated visual objects. Therefore it seems likely that incorporating image segmentation techniques into the process of sound source separation might be beneficial. This is especially appealing, because the previously introduced methods can be directly applied in the auditory domain without any modifications.

Another link between both domains is that image segmentation, as well as models of

auditory sensation, can be associated with the *Gestalt principles*, described in Bregman (1990). Hence, incorporating image segmentation techniques combined with CASA-inspired processing based on *Gestalt principles* provides an opportunity for further research into this direction.

4.5.2 High-level scene analysis using semantic labels

In contrast to the aforementioned approach of using GMs to solve low-level tasks like segmentation of auditory cues, probabilistic methods also provide capabilities for high-level analysis of acoustic scenes. This shall be illustrated here with a simple example from the field of CASA. In this example, two different acoustic scenes are considered, which are denoted as "traffic" and "car accident". To keep the scenario description simple, four different sound classes are present in these scenes: a yelling person, the sound of a car crashing, engine sounds and footsteps. A Bayesian network (BN) that captures the sounds present in this scenario is depicted in Fig. 4.11. The degree of how likely a specific sound occurs in either a "traffic" or "accident" scene is modeled via the depicted conditional probability tables (CPTs). For instance, the event of a person yelling is more likely to be observed if a car accident has actually happened. Similarly, footsteps might be present in both scenes, even though, their occurrence might be less likely if an accident happened, due to the fact that pedestrians might be stopped from crossing the scene by police forces.



	Y	
S	T	F
traffic	0.10	0.90
accident	0.75	0.25

	C	
S	T	F
traffic	0.05	0.95
accident	0.99	0.01

	E	
S	T	F
traffic	0.90	0.10
accident	0.25	0.75

	F	
S	T	F
traffic	0.80	0.20
accident	0.40	0.60

Figure 4.11: Example of a BN describing the semantic relationships of sounds to acoustic scenes in a simple traffic/car accident scenario. The variable S can be set boolean to either "traffic" or "accident". The variables Y, C, E, F describe whether the audio events "yelling", "crash", "engine", or "footsteps" are present in the auditory scene.

It must be noted that the CPTs in this particular example are "hand-crafted" solely by intuition. In order to design BNs that capture a specific scenario in a more realistic fashion, CPTs could also be trained from data.

Drawing Samples from a Bayesian Network. The simple BN introduced here is already able to generate a large variety of acoustic scenes, given only the structure of the network and the corresponding CPTs. This is achieved by drawing samples from the network. A sample can be considered as a specific instantiation of the BN.

Samples can either be drawn from the joint probability $P(S, Y, C, E, F)$ or from conditional probabilities if certain prior observations are assumed. For example, samples for a particular scene could be generated by setting S to a specific instance before the sampling process. If $P(S = \text{traffic})$ is fixed a priori, the BN will generate samples according to $P(Y, C, E, F \mid S = \text{traffic})$, hence the sound classes "engine" and "footsteps" will more likely occur in the generated scene than "yelling" or "crash". Sampling BNs can be performed efficiently by exploiting the fundamental assumptions on independence properties between random variables in the network. Specifically, it is possible to sample each variable separately if an instantiation of the variable's parents is provided. This is the case when setting $P(S = \text{traffic})$. Because the remaining variables are independent, given their mutual parent S, they can be sampled individually according to the corresponding entries in their CPTs

$$y \sim P(Y, \mid S = \text{traffic}) = P(0.10,\, 0.90) \tag{4.18}$$
$$c \sim P(C, \mid S = \text{traffic}) = P(0.05,\, 0.95) \tag{4.19}$$
$$e \sim P(E, \mid S = \text{traffic}) = P(0.90,\, 0.10) \tag{4.20}$$
$$f \sim P(F, \mid S = \text{traffic}) = P(0.80,\, 0.20) \tag{4.21}$$

where y, c, e and f are specific instances of the RVs. The probability distributions (4.18)–(4.21) can be assumed as Bernoulli distributions, each having the probability mass function

$$P(X = x \mid k, l) = \begin{cases} k = (1 - l), & \text{for } x = \text{T} \\ l, & \text{for } x = \text{F} \end{cases}$$

where X is the RV and x is an instantiation of this RV. Sampling from a Bernoulli distribution is a process that can be implemented easily and efficiently. A basic implementation is outlined in Alg. 1. By applying this to the BN considered here, it is possible to generate arbitrary "traffic" and "accident" scenes by drawing samples from the corresponding conditional probabilities. For instance, sampling the BN depicted in Fig. 4.11, given the prior observation $P(S = \text{traffic})$ might provide $y = \text{F}$, $c = \text{F}$, $e = \text{T}$ and $f = \text{T}$ as an output. These samples could then be used to randomly pick corresponding sounds from a sound database to automatically generate a scene. Hence, treating the problem of acoustic scene

generation as drawing samples from a BN provides a versatile means to generate arbitrary scenes given a semantic context of associated sound sources.

Automatic Generation of Complex Acoustic Scenes. It was shown in the previous section that automatic generation of acoustic scenes can be achieved by modeling the conditions of a scenario with a BN. Nevertheless, for the sake of simplicity, a very basic scenario containing four different sounds was introduced as an example. However, following the paradigm of using BNs in this context, more complex scenes can be generated via simple extensions of the underlying GM. This can be achieved by integrating additional RVs to the network structure and discarding the constraint of the network being solely discrete. Without going into detail, Tab. 4.5 lists some initial ideas on RVs that might allow for improving the amount of realism in a generated scene. It is obvious that by introducing physical properties like reverberation time into the scenario model, the corresponding RVs can not be discrete anymore, but rather continuous probability distributions will have to be assumed. This results in a hybrid BN, containing discrete, as well as continuous nodes, which makes sampling, learning, and inference procedures computationally more demanding. Nevertheless, powerful methods exist for approximately solving these tasks.

Learning the network parameters. As was previously discussed, it might be beneficial to learn CPTs, and distribution parameters from data, rather than "hand-crafting" a BN for a specific scenario. However, this task can be quite challenging, depending on the size of the network and the amount of realism that should be achieved. In any case, it is of major importance to gather an amount of training data that is large enough to avoid overfitting the network during the training process. Essentially, there are several possibilities for acquiring training data, which will be briefly introduced here.

Considering the previous example of a "traffic" or "accident" scenario, training data can be gathered by data mining techniques. For instance, the prior probability of the scene $P(S)$

Algorithm 1 Drawing samples from a Bernoulli distribution

```
function SampleBernoulli(k)
Sample a random number u ∈ [0, 1] from uniform distribution 𝒰
    if u ≤ 1 − k then
        return false
    else
        return true
    end if
end function
```

Random variable	Type	Possible distributions
Number of sound sources	Discrete	Binomial
Number of loudspeakers	Discrete	Binomial
Listener position	Continuous	Uniform, Gaussian, Gaussian Mixture
Source position	Continuous	Uniform, Gaussian, Gaussian Mixture
Loudspeaker position	Continuous	Gaussian
Source loudness	Continuous	Gaussian
Reverberation time	Continuous	Inverse Gaussian, Inverse Gamma
Room volume	Continuous	Gamma, Rice
Absorption coefficients	Continuous	Uniform, Beta

Table 4.5: This table shows RVs that could be used for extending a scenario model based on BNs. The examples shown here can be used within both the DASA and the QoE application scenarios.

represents how likely it is that a car accident occurs in a normal traffic situation. Intuitively, the probability for $P(S = \text{accident})$ should be relatively small, hence it was set to 0.05. Considering a realistic scenario, this probability might even be much smaller than considered here. It might also depend on additional factors that were not taken into account in the example, e.g. speed limits, traffic volume, road conditions, weather etc. Integrating all these additional variables would increase realism, but likewise also increase the complexity of the model. A possible trade-off could be achieved by fixing the environmental conditions to a certain degree and try to gather information on how likely it is that an accident occurs given these conditions. For example, the scenario may be provided with an additional RV for the weather, denoted by W, which can take the values "sunny", "rain" or "snowing". As described before, the weather will have a direct effect on the possibility that a car accident might occur in the given scenario. In order to train the corresponding CPTs, data has to be gathered from appropriate sources. In this case, this might be traffic statistics available on the internet or in specific scientific publications. The process of gathering data could also be automated by using data-mining techniques or by retrieving and integrating expert knowledge into the system.

For the DASA and QoE application scenarios considered in Two!Ears, the aforementioned approach can be used to gather data for many parameters of interest, e.g.:

- Sound pressure of sources that should be modelled in a particular scenario. Lists and tables containing measured or averaged values can be found online or in the specific literature.
- Average reverberation time in different environments (domestic, street, church, etc.). This data could be gathered from lists and tables related to acoustic measurements.
- Absorption coefficients for different wall, floor, and ceiling types. Sources for data

acquisition might be the same as for the acquisition of reverberation time.

As can be seen from these examples, it appears that data mining approaches might be primarily suited for collecting datasets that represent physical attributes of an acoustic scene. In contrast, information concerning the semantic structure of a particular scenario can not be gathered by simply harvesting all available information sources. Here, entities in a scene might already be prone to subjective interpretation. Therefore, data acquisition based on expert knowledge might be more suitable.

Integration into the Two!Ears framework. The approach of using GMs for a high-level analysis of acoustic scenes introduced in Sec. 4.5.2 provides a flexible framework to be used in conjunction with lower-level processing components. In this case, semantic labels generated by e.g. source type classifiers or localisation modules, would serve as observations for the BN that describes specific properties of the acoustic scene. The remaining variables could then be inferred via approximate inference techniques, given the observations. Questions to be addressed in future investigations mainly regard possible ways of interconnecting higher-level inference stages with low level processing modules within the blackboard system. The current Two!Ears software system provides a flexible basis for this integration, which will be extended accordingly in the future.

5 Case study

This chapter presents a case study that demonstrates the operation of the TWO!EARS system of a specific scenario. The system is evaluated on a scenario in which a female voice must be located in the presence of male voice maskers, under conditions in which the masker positions are unknown or known *a priori*. The task therefore requires segmentation, source localisation, gender recognition and top-down feedback. The scenario is described first, and then each of these aspects of the system is described in turn.

5.1 Scenario DASA-1

The scenario we chose for evaluation (denoted DASA-1; see also Deliverable D6.1.2) is based on one used in a psychophysical study by Kopčo *et al.* (2010). In their study, listeners were presented with speech stimuli via an array of 11 loudspeakers, in a small room (approximately $3 \times 5 \times 2.5$ m). The task for the listener was to localise a female voice, which was presented concurrently with four male-voice maskers. The experimental setup is shown in Figure 5.1. The loudspeakers were arranged in an arc of radius 1.5 m in front of the listener, with an angular separation of 10° between adjacent speakers.

Female and male speech was drawn from a small corpus of monosyllabic words (Kidd *et al.*, 2008). The male maskers were presented in one of five patterns, as shown in panel B of Figure 5.1. The female target voice could be presented from any of the 11 loudspeakers (including a speaker that was also emitting a masker sound). Trials were presented in one of two conditions. In the *fixed* condition, the listener was cued to the location of the masker sounds by hearing the phrase "fixed maskers" from each masker position in turn. A number of trials were then presented in which the masker pattern was kept fixed, and the target location was varied randomly from trial-to-trial. In the *mixed* condition, the target and masker pattern were both varied randomly from trial-to-trial.

Kopčo *et al.* (2010) measured the ability of listeners to localise the target source. They found that the RMS error in subjects' localisation judgments was reduced by 20% when the locations of the masker sounds were known *a priori*. This effect was strongest when the target position did not coincide with any of the masker positions (36% reduction in error rate). Their findings therefore suggests that listeners' expectations of the spatial arrangement of

sound sources influences their ability to localise a target sound, a finding which is consistent with the feedback pathways hypothesised in the TWO!EARS system.

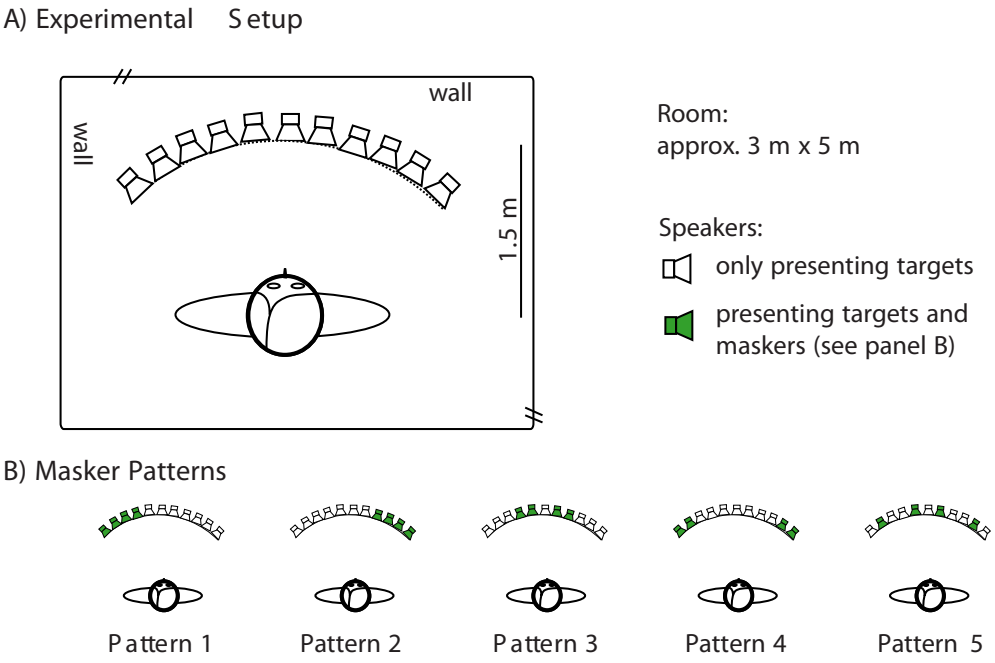


Figure 5.1: Experimental setup and masker patterns in the study of Kopčo *et al.* (2010). In the experimental setup (A), eleven loudspeakers are evenly spaced in a 1.5m arc in front of the listener (10° separation between speakers). Four male speech maskers (B) were presented in one of five patterns. The maskers were presented concurrently with a female speech target [Figure reproduced from Kopčo et al. (2010) Speech localisation in a multitalker mixture, J. Acoust. Soc. Am. 127 (3), p. 1451 with the permission of AIP Publications].

For the simulations reported here, the stimuli used in the Kopčo *et al.* (2010) experiment were reproduced via the TWO!EARS binaural simulator. This spatialised sound sources using a set of anechoic HRIRs combined with a 'shoebox' model of room acoustics, configured to reproduce the room dimensions, listener position and loudspeaker positions used by Kopčo *et al.* (2010). The speech stimuli were those used in the original experiment[1].

1 Our thanks to Norbert Kopčo and Peter Toth for making these signals available to us.

5.2 System description

5.2.1 Source localisation

The aim of sound localisation for this scenario is to estimate the posterior probability of a source being present for each azimuth angle. Since there are five speakers present at the same time, the estimated posterior distribution would exhibit high probabilities at multiple azimuth angles.

We adopt the DNN-based machine-hearing system for robust localisation of multiple speakers in reverberant conditions (Ma *et al.* (2015b), Section 4.1.1). An auditory front-end was employed to analyse binaural ear signals, consisting of a bank of 32 overlapping Gammatone filters with centre frequencies uniformly spaced on the equivalent rectangular bandwidth (ERB) scale between 80 Hz and 8 kHz (Wang and Brown, 2006). Inner hair cell function was approximated by half-wave rectification. Afterwards, the cross-correlation between the right and left ears was computed independently for each frequency channel using overlapping frames of 20 ms duration with a shift of 10 ms.

The cross-correlation function with a lag range of ± 1.1 ms produced a 37-dimensional binaural feature space for each frequency channel. This was supplemented by the ILD, forming a final 38-dimensional (38D) feature vector. DNNs were then used to map the 38D binaural feature set to corresponding azimuth angles. A separate DNN was trained for each frequency channel. The DNN consists of an input layer, 4 hidden layers, and an output layer. The input layer contained 38 nodes and each node was assumed to be a Gaussian random variable with zero mean and unit variance. Therefore the 38D binaural feature input for each frequency channel was first Gaussian normalised, before being fed into the DNN. The hidden layers had sigmoid activation functions, and each layer contained 128 hidden nodes. The number of hidden nodes was heuristically selected as more hidden nodes add more computation and did not improve localisation accuracy in this study. The output layer contained 21 nodes corresponding to the 21 azimuth angles in the range of [-50 50] degrees (5 deg steps), which includes the 11 target azimuth angles considered in this study. The "softmax" activation function was applied at the output layer.

Given the observed feature set $\vec{x}_{t,f}$ at time frame t and frequency channel f, the 21 "softmax" output values from the DNN for frequency channel f were considered as posterior probabilities $\mathcal{P}(k|\vec{x}_{t,f})$, where k is the azimuth angle and $\sum_k \mathcal{P}(k|\vec{x}_{t,f}) = 1$. The posteriors were then integrated across frequency and time to yield the posterior probabilities for each azimuth considered.

Figure 5.2 shows a few examples of the system output for the DASA-1 scenario. Various masker patterns and target positions are included. In each panel, the reference target position is marked by a blue circle and the reference masker positions are marked by red

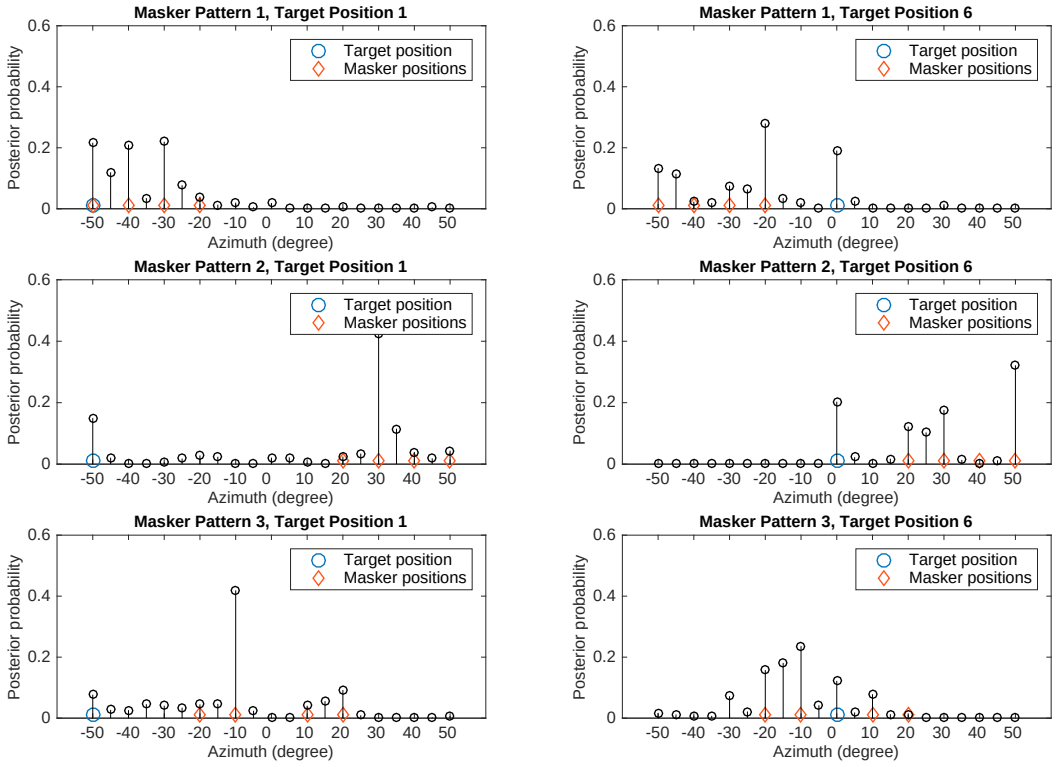


Figure 5.2: Posterior probabilities of source azimuth for various target/masker patterns.

diamonds. It can be seen that in most cases, the system produced a high probability at the target azimuth and it is among the five highest probabilities. However, it is not clear which azimuth corresponds to the target speaker without knowledge about the target speaker. In the next sections we will discuss the use of further knowledge available for this scenario, e.g. the gender and spectral characteristics of the target speaker. Such knowledge can then be used to perform segmentation of the target speaker from the maskers and inform the correct target azimuth.

5.2.2 Segmentation

The purpose of sound source segmentation is to assign specific time-frequency units of the auditory feature space to individual sound sources. This can either be done using a hard-assignment via binary masking or by computing soft-masks, which yields a smooth segmentation of the feature space. The latter can also be interpreted as a probabilistic assignment of time-frequency units to sound sources. The segmentation process used in this case study focuses on estimating soft-masks, which fits the general approach of using

probabilistic models throughout the whole framework.

Feature extraction and supervised training. The proposed segmentation framework is embedded into a KS, which is integrated into the blackboard system. It is triggered each time a signal block of length T_B has been acquired by the auditory front-end (AFE). For each block, the KS takes as inputs a set of angular positions

$$\mathcal{P} = \Big\{\theta_1, \ldots, \theta_P \ \in \ [-\pi, \pi]\Big\}, \tag{5.1}$$

which are estimated using the localisation KS described in Sec. 5.2.1. Additionally, a time-frequency map of interaural cross-correlation (ICC) and ILD features is computed by the AFE. Within this map, each time-frequency bin at time-index n and frequency index m is associated with a feature vector

$$\boldsymbol{x}_{nm} = \Big(\rho_{nm}^{(1)}, \ldots, \rho_{nm}^{(L)}, \delta_{nm}\Big)^T, \tag{5.2}$$

where $\rho_{nm}^{(l)},\ l = 1, \ldots, L$ is the ICC coefficient at time-lag index l and δ_{nm} is the ILD. Hence, each feature vector comprises $L + 1$ dimensions, where L is the number of ICC coefficients. This feature representation is subsequently used to derive a map of estimated angular positions ϕ_{nm} for each time-frequency unit of the signal block. Therefore, a mapping function $f : \ \mathbb{R}^{L+1} \mapsto \mathbb{R}$ is learned via supervised training of a regression ν-support vector machine (SVM) introduced in Schölkopf *et al.* (2000) for each frequency index m. This yields a set of M SVMs, able to predict the angular position at each time-frequency unit as

$$\phi_{nm} = \sum_i \beta_m^{(i)} k(\boldsymbol{x}_{nm}, \boldsymbol{x}_m^{(i)}) + b_m, \tag{5.3}$$

where $\beta_m^{(i)}$ and b_m are the parameters of the m-th SVM, $k(\boldsymbol{x}_{nm}, \boldsymbol{x}_m^{(i)})$ is a kernel function and $\boldsymbol{x}_m^{(i)}$ is the i-th feature vector of the training set. All SVMs are trained on features derived from head-related impulse responses (HRIRs) of the Knowles Electronics manikin for acoustic research (KEMAR) dummy head presented in Wierstorf *et al.* (2011). White noise is used as an input signal throughout the entire training phase. A linear kernel is applied in this case study, although the evaluation of nonlinear kernels is planned for future experiments.

Segmentation via spatial clustering. During execution of the KS, Eq. (5.3) is used to assign estimates of angular positions to each time-frequency unit of the acquired signal block. This representation serves as the basis for a subsequent clustering step. However, conventional clustering techniques like k-means (MacQueen (1967)) or Gaussian mixture models (GMMs) (Dempster *et al.* (1977)) might not be suitable for the problem at hand.

This is because the available observations are azimuth angles, hence they originate from a circular probability distribution bounded in $[-\pi, \pi]$.

Therefore, an alternative clustering technique is applied here, which is based on a mixture of von Mises distributions introduced in Banerjee *et al.* (2005). The von Mises distribution is defined as

$$\mathcal{VM}(\phi, \,|\, \mu, \kappa) = \frac{1}{2\pi I_0(\kappa)} \exp\Big\{\kappa \cos(\phi - \mu)\Big\}, \tag{5.4}$$

where $\phi \in [-\pi, \pi]$ is an angle, μ is the circular mean, κ is the concentration parameter and $I_0(\,\cdot\,)$ is the modified Bessel function of order 0. Subsequently, the probabilty density function (PDF) of a mixture of von Mises distributions can be derived using Eq. (5.4) as

$$p(\phi \,|\, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\kappa}) = \sum_{k=1}^{K} \pi_k \mathcal{VM}(\phi \,|\, \mu_k, \kappa_k) \tag{5.5}$$

with

$$\boldsymbol{\pi} = [\pi_1, \ldots, \pi_K]^T, \quad \boldsymbol{\mu} = [\mu_1, \ldots, \mu_K]^T, \quad \boldsymbol{\kappa} = [\kappa_1, \ldots, \kappa_K]^T,$$

where K is the number of mixture components and π_k is the mixture weight of the k-th component, satisfying $\sum_{k=1}^{K} \pi_k = 1$. The segmentation process is based on the assumption that the number of active sound sources S is known a priori. Therefore, the number of mixture components in model (5.5) is set to $K = S + 1$, so that S components correspond to the angular positions of the individual sound sources. The remaining K-th mixture component is used to estimate potential background noise. By assuming an ideally diffuse noise field, a circular uniform distribution can be assumed for the corresponding angular positions. Within the proposed framework, this can be modelled by setting the concentration parameter of the associated mixture component to zero. Furthermore, using the $P \leq S$ estimates of potential source positions provided via the blackboard (5.1), the circular means of P additional mixture components are fixed to these positions. For a given map of estimated source positions

$$\boldsymbol{P} = \begin{bmatrix} \phi_{11} & \cdots & \phi_{1M} \\ \vdots & \ddots & \vdots \\ \phi_{N1} & \cdots & \phi_{NM} \end{bmatrix},$$

a vector of observations is derived by changing the double element index nm to a single index $i = 1, \ldots, N_\mathrm{s}$ with $N_\mathrm{s} = N \cdot M$ over all time and frequency units and stacking the columns of $\boldsymbol{P}$ according to

$$\begin{aligned} \boldsymbol{\phi} = \mathrm{vec}(\boldsymbol{P}) &= [\phi_{11}, \ldots, \phi_{N1}, \phi_{12}, \ldots, \phi_{N2}, \ldots, \phi_{1M}, \ldots, \phi_{NM}]^T \\ &= [\phi_1, \ldots, \phi_i, \ldots, \phi_{N_\mathrm{s}}]^T. \end{aligned} \tag{5.6}$$

This allows to express the log-likelihood of Eq. (5.5) given the observations $\boldsymbol{\phi}$ as

$$\mathcal{L}(\boldsymbol{\phi} \,|\, \boldsymbol{\pi},\, \boldsymbol{\mu},\, \boldsymbol{\kappa}) = \sum_{i=1}^{N_\mathrm{s}} \log\Big(\pi_K \frac{1}{2\pi} + \sum_{k=1}^{K-1} \pi_k \mathcal{VM}(\phi_i \,|\, \mu_k,\, \kappa_k)\Big), \tag{5.7}$$

with

$$\mu_k = \theta_k,\ \forall\, k \leq P.$$

The remaining parameters of Eq. (5.7) are estimated using an expectation-maximisation (EM) scheme based on the approach presented by Hung *et al.* (2012). The parameter estimates at each maximisation step are given as

$$\mu_k = \begin{cases} \theta_k, & \text{if } k \leq P \\ \operatorname{atan2}\Big(\sum_{i=1}^{N_\mathrm{s}} \gamma_{ik} \sin(\phi_i),\, \sum_{i=1}^{N_\mathrm{s}} \gamma_{ik} \cos(\phi_i)\Big), & \text{if } k > P \\ 0, & \text{if } k = K \end{cases}, \tag{5.8}$$

$$\kappa_k = \begin{cases} A^{-1}\Big(\frac{\sum_{i=1}^{N_\mathrm{s}} \gamma_{ik} \cos(\phi_i - \mu_k)}{\sum_{i=1}^{N_\mathrm{s}} \gamma_{ik}}\Big), & \text{if } k \leq K \\ 0, & \text{if } k = K \end{cases} \tag{5.9}$$

$$\pi_k = \frac{1}{N_\mathrm{s}} \sum_{i=1}^{N_\mathrm{s}} \gamma_{ik}, \tag{5.10}$$

with

$$\gamma_{ik} = \frac{\pi_k \mathcal{VM}(\phi_i \,|\, \mu_k,\, \kappa_k)}{\sum_{j=1}^{K} \pi_j \mathcal{VM}(\phi_i \,|\, \mu_j,\, \kappa_j)} \tag{5.11}$$

and

$$A(x) = \frac{I_1(x)}{I_0(x)}. \tag{5.12}$$

Estimating the concentration parameters κ_k requires inverting the function given in Eq. (5.12). This problem cannot be solved analytically, therefore the inverse function has to be approximated. In this case study, the approximation scheme introduced by Best and Fisher (1981) is applied to estimate the concentration parameters. The EM algorithm utilises Eqs. (5.8)–(5.12) to incrementally update the parameter estimates during the optimisation process. The initial model parameters are computed using the circular k-means algorithm described in Banerjee *et al.* (2005). An outline of the complete method used in this case study is summarised in algorithm 2.

Soft mask computation. Following the parameter estimation procedure, the model (5.5) can be used to derive soft masks for the active sound sources. Each soft mask is specified

Algorithm 2 EM for circular clustering

Inputs:

- Number of active sound sources S
- Estimated source positions for current signal block (5.1)
- Estimated azimuth angles for all time-frequency units of the current block (5.6)

Initialisation: Run circular k-means to get initial parameters π_k, μ_k, κ_k and γ_{ik}
repeat
  E-Step:
    Compute responsibilities γ_{ik} using Eq. (5.11)
  M-Step:
    Re-estimate circular means μ_k using Eq. (5.8)
    Re-estimate concentration parameters κ_k using Eqs. (5.9) and (5.12)
    Re-estimate mixture proportions π_k using Eq. (5.10)
  Evaluate the log-likelihood using Eq. (5.7)
until log-likelihood $\mathcal{L}(\boldsymbol{\phi} \,|\, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\kappa})$ converges

as an $N \times M$ matrix

$$\boldsymbol{M}^{(i)} = \begin{bmatrix} m_{11}^{(i)} & \cdots & m_{1M}^{(i)} \\ \vdots & \ddots & \vdots \\ m_{N1}^{(i)} & \cdots & m_{NM}^{(i)} \end{bmatrix}, \tag{5.13}$$

where $i = 1, \ldots, S$ is the sound source index. The individual masking coefficients at each time-frequency unit are computed by evaluating the likelihood of the individual mixture components of the model (5.5), given the estimated azimuth angle (5.3), which yields

$$m_{nm}^{(i)} = \frac{\pi_i \mathcal{VM}(\phi_{nm} \,|\, \mu_i, \kappa_i)}{\sum_{j=1}^{K} \pi_j \mathcal{VM}(\phi_{nm} \,|\, \mu_j, \kappa_j)}.$$

Additionally, it is possible to derive a soft mask of the background noise by setting $i = K$. An example of soft masks that were generated by the proposed framework is depicted in Fig. 5.3.

Future developments. Besides using nonlinear kernels for the SVM regression stage, future developments of the proposed segmentation framework will focus on improving robustness in the presence of reverberation and dealing with dynamic scenes. Furthermore, the framework provides several possibilities for including feedback mechanisms. This comprises feedback through motion, like head and translatory movements, but also options for adaptation, e.g. by changing specific filter bandwidths or by adaptively suppressing unreliable channels. Additionally, top-down knowledge of specific source types which are present in a scene can be used to further improve the segmentation performance.

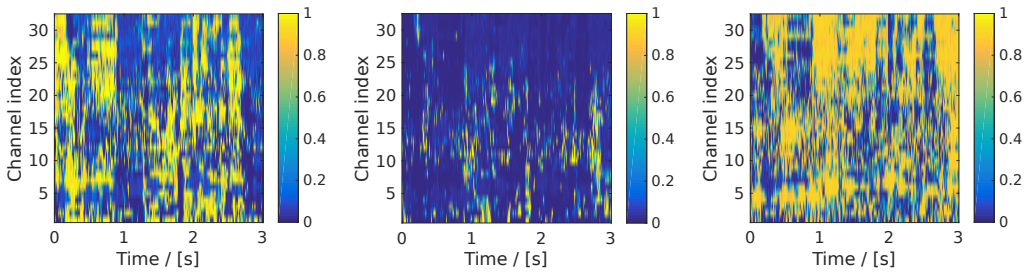


Figure 5.3: Soft masks, derived from a binaural mixture containing three speech sources at $-30°$ (left image), $30°$ (centre image) and $0°$ (right image).

5.2.3 Gender recognition

The considered case study requires that the gender of all speakers present in the scene can be estimated for each speaker individually. Therefore, a KS for gender recognition has been developed, which is able to work with segmented auditory features provided by the segmentation KS described in Sec. 5.2.2.

Auditory features for gender recognition. A prominent feature for gender recognition tasks is the pitch of the speakers voice, see e.g. Lakra *et al.* (2013). However, deriving pitch features from segmented auditory cues in a multi-talker environment proves to be impractical in most cases. This is due to the fact that pitch features are commonly associated with a specific time-frame, whereas segmented sets of auditory features are based on individual time-frequency units. Hence, a different type of auditory feature is considered in this case study, namely *formant maps*.

Formant maps represent the formants of an audio signal in time and frequency. They are derived from a ratemap representation of a monaural signal

$$\boldsymbol{R} = \begin{bmatrix} r_{1,1} & \cdots & r_{1,M} \\ \vdots & \ddots & \vdots \\ r_{N,1} & \cdots & r_{N,M} \end{bmatrix}, \tag{5.14}$$

where each element r_{nm} depicts the amplitude of the ratemap at frame index n and frequency index m. As formants correspond to frequency regions with concentrated high signal energy, a formant map can be derived from Eq. (5.14) by computing the derivative

at each point along the frequency axis via

$$f_{nm} = \begin{cases} r_{n,\,m+1} - r_{n,\,m}, & \text{if } m = 1 \\ r_{n,\,m} - r_{n,\,m-1}, & \text{if } m = M \\ \frac{1}{2}\Big(r_{n,\,m+1} - r_{n,\,m-1}\Big), & \text{otherwise} \end{cases}. \tag{5.15}$$

This yields a time-frequency representation of the formants which can be processed by soft-masks that have been estimated during the segmentation process. An example of a formant map derived from auditory features of a speech signal is depicted in Fig. 5.4.

The analysis of formant maps reveals that distinct peaks, which mainly occur in the lower frequency ranges, are systematically shifted towards higher frequencies for female speakers in comparison to speech signals from male speakers. This behaviour is explains the discriminative power of formant maps in the context of gender recognition. It should be noted, that more suitable representations of formant map features can be derived for this task. Especially in higher frequencies, not much discriminative information is present, hence, an appropriate mechanism to reduce the feature dimension could be considered. Additionally, the height of formant peaks depends on the level of the signal in the corresponding frequency region. Therefore, future investigations in the context of gender recognition will focus on increasing robustness by discarding irrelevant frequency regions and making formant maps invariant to the signal level. For the case study presented here, unmodified formant maps according to Eq. (5.15) are used.

Classifiers for gender recognition. As the gender recognition KS is working on segmented auditory features, a classifier that is capable of handling this specific feature representation is required. For this case study, a probabilistic classification scheme based on quadratic discriminant analysis (QDA) as described in (Hastie *et al.*, 2001, p. 106–119) is considered. In QDA, the probability of an observation vector $\boldsymbol{x}_n = [f_{n,\,1}, \ldots, f_{n,\,M}]^T$ at time frame n, given a class c_i, $i \in \{\text{male, female}\}$ is specified by a multivariate Gaussian distribution

$$p(\boldsymbol{x}_n \,|\, c_i) = \frac{1}{\sqrt{(2\pi)^M |\boldsymbol{\Sigma}_i|}} \exp\Big(-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_i)\Big), \tag{5.16}$$

where M is the number of filterbank channels and $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are the mean vector and covariance matrix of the i-th class, respectively. Applying Bayes' rule on Eq. (5.16) yields

$$p(c_i \,|\, \boldsymbol{x}_n) = \frac{p(\boldsymbol{x}_n \,|\, c_i) p(c_i)}{p(\boldsymbol{x}_n)}. \tag{5.17}$$

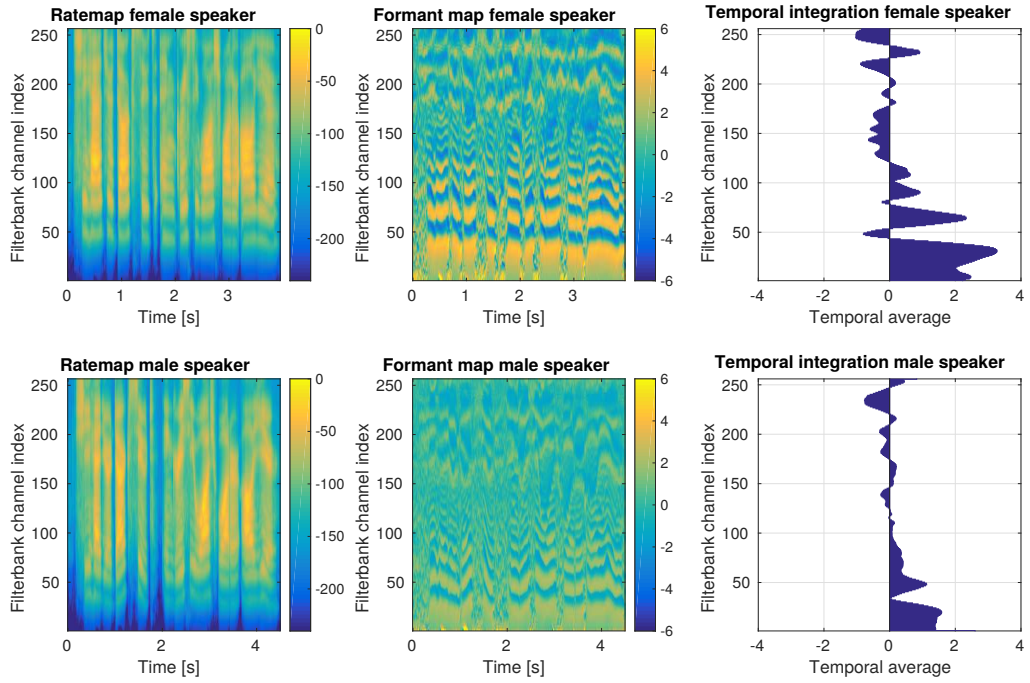


Figure 5.4: Ratemaps, formant maps and temporal averages of formant maps (from left to right) for a female and a male speaker. The temporal averages show two distinct peaks in the lower frequency range, with a systematic shift towards higher frequencies for the female speaker. A gammatone filterbank with $M = 256$ channels was used, in order to produce maps with high resolution.

By taking the logarithm of Eq. (5.17), the log-likelihood of class c_i given the observation $\boldsymbol{x}_n$ can be derived as

$$\mathcal{L}(c_i \,|\, \boldsymbol{x}_n) = -\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_i) - \frac{1}{2}\log(|\boldsymbol{\Sigma}_i|) + \log(c_i) + \text{const.}, \tag{5.18}$$

where the term $\log(c_i)$ corresponds to the prior probability of the i-th class. The distribution parameters $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are computed individually for each class, using formant map features of the male and female stimuli provided by the database described in Kopčo *et al.* (2010). The training was conducted on undisturbed features in anechoic conditions.

In order to deal with disturbances introduced by the segmentation process during testing, a simple uncertainty-of-observation technique is used in this study. At each time-frame, a set of masking coefficients $\boldsymbol{m}_n^{(j)} = [m_{n,1}^{(j)}, \ldots, m_{n,M}^{(j)}]$ for the j-th source present in the scene is estimated by the segmentation KS. The masked format map features are subsequently

derived via a masking matrix

$$\boldsymbol{W}_n^{(j)} = \mathrm{diag}(m_{n,1}^{(j)}, \ldots, m_{n,M}^{(j)}), \tag{5.19}$$

which allows to compute the masked features corresponding to the j-th source as

$$\tilde{\boldsymbol{x}}_n^{(j)} = \boldsymbol{W}_n^{(j)} \boldsymbol{x}_n^{(j)}. \tag{5.20}$$

The most likely class membership of the resulting masked frame (5.20) is then derived via a maximum likelihood approach according to

$$\begin{aligned} \hat{c}_n^{(j)} = \arg\max_i \Big[-\frac{1}{2}(\tilde{\boldsymbol{x}}_n - \boldsymbol{W}_n^{(j)}\boldsymbol{\mu}_i)^T (\boldsymbol{W}_n^{(j)}\boldsymbol{\Sigma}_i(\boldsymbol{W}_n^{(j)})^T)^{-1}(\tilde{\boldsymbol{x}}_n - \boldsymbol{W}_n^{(j)}\boldsymbol{\mu}_i) \\ - \frac{1}{2}\log(|\boldsymbol{W}_n^{(j)}\boldsymbol{\Sigma}_i(\boldsymbol{W}_n^{(j)})^T|) + \log(c_i)\Big]. \end{aligned} \tag{5.21}$$

The final decision of the class membership for a specific source is based on a majority vote over a set of frame-wise decisions $\hat{c}_n^{(j)}$, $n = 1, \ldots, N$, where N is the number of frames of the signal block that should be analysed.

5.2.4 Top-down feedback

Human listeners must answer two questions in order to fully understand an acoustic scene; *what* the sound sources are, and *where* they are. In machine hearing, these two issues have been addressed by many studies via computational approaches for sound source separation, classification and localisation (Wang and Brown, 2006). Recently, an approach was developed for binaural localisation that exploits top-down knowledge about the source spectral characteristics in the acoustic scene Ma *et al.* (2015a). Here we discuss application this approach to the DASA-1 scenario. More details of the method can be found in Deliverable Report D4.2.

We adopt the same DNN-based localisation system as reported in Section 5.2.1. For each of the 32 frequency channels, a DNN is used to map binaural features (obtained from a cross-correlogram) to 21 azimuth angles in the range of [-50 50] degrees (5 deg steps), which includes the 11 target azimuth angles considered in this study.

When the posteriors from the DNNs for each frequency channels are integrated, we introduce ω_{tf} as a weighting factor between $[0, 1]$:

$$P(\phi|\vec{o}_t) = \frac{\prod_f P(\phi|\vec{o}_{tf})^{\omega_{tf}}}{P(\vec{o}_t)}, \tag{5.22}$$

where

$$P(\vec{o}_t) = \sum\nolimits_\phi \prod\nolimits_f P(\phi|\vec{o}_{tf})^{\omega_{tf}}. \tag{5.23}$$

Here ω_{tf} is used to selectively weight the contribution of binaural cues from each time-frequency bin in order to localise the attended target source in the presence of competing sources. This allows cues that derive from a frequency channel dominated by the target source to be emphasised; or conversely, cues that derive from an interfering source can be penalised. In this study top-down information from source models is combined to jointly estimate these localisation weights.

Source spectral characteristics were modelled using ratemap features Brown and Cooke (1994). A ratemap is a spectro-temporal representation of auditory nerve firing rate, extracted from the inner hair cell output of each frequency channel by leaky integration and downsampling. For the binaural signals used here, the ratemap features were computed for each ear and then averaged across the two ears. They were finally log-compressed to form 32D feature vectors $\vec{x}_t$.

Let λ_s represent the spectral characteristics of a sound source s in a set of source models $s = 1, \ldots, \mathcal{S}$. The set of source models are employed to jointly explain the observed ratemap features. In particular, given the observed log-compressed ratemap feature vector $\vec{y}_t = [y_{t1}, \ldots, y_{t32}]^\top$ extracted at time frame t from the binaural signals, we want to determine whether each feature y_{tf} is dominated by the energy of the target source x_{tf} or corrupted by the combined energy of interfering sources n_{tf}. Under the *log-max* approximation Varga and Moore (1990) of the interaction function between two acoustic sources, i.e. $y_{tf} \approx \max(x_{tf}, n_{tf})$, the localisation weight ω_{tf} can be defined as the probability of y_{tf} being dominated by x_{tf}

$$\omega_{tf} = P(x_{tf} = y_{tf}, n_{tf} \leq y_{tf} | \vec{y}_t, \lambda_x, \lambda_n), \tag{5.24}$$

where λ_x and λ_n are the models for the target and interfering sources, respectively. Here, the source models are represented as GMMs with diagonal covariance matrices. See Ma *et al.* (2015a) for full details of the derivation.

Figure 5.5 shows the system output for the same examples used in Figure 5.2 for the DASA-1 scenario. In each panel, the reference target location is marked by a blue circle and the reference masker locations are marked by red diamonds. It can be seen that with the top-down knowledge from the source spectral models, the system now produced the highest probability at the correct target azimuth in most cases.

When the locations of the masker speakers are cued prior to each listening trial (the 'Fixed' setup), the system can also make use of such prior knowledge by penalising cues that derive from the maskers. This can be done by setting the posterior probabilities at the reference masker azimuths to zero. However, since the target speaker and the masker speakers can be co-located, simply setting the probabilities at the masker azimuths to zero will also inhibit the probability of the correct target azimuth. We therefore only apply the penalty at the reference masker azimuths in the frequency channels that are more likely to be

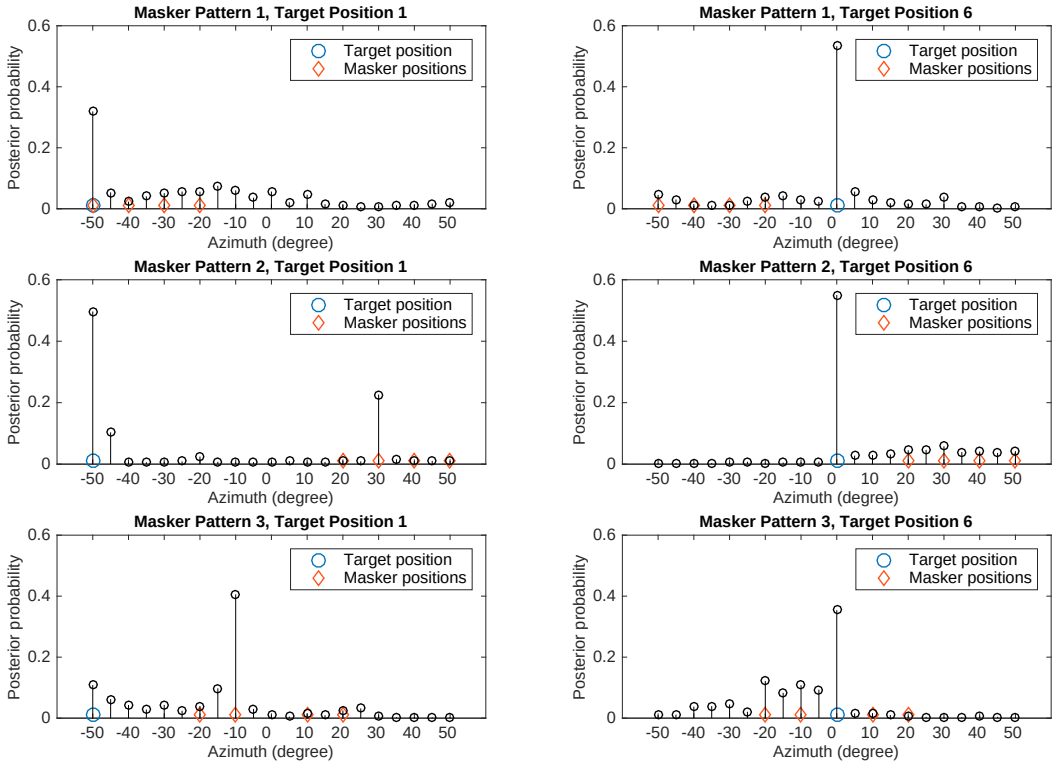


Figure 5.5: Posterior probabilities of source azimuth for various target/masker patterns with top-down feedback.

dominated by the masking speaker, i.e. where the estimated weight ω_{tf} is smaller than 0.5.

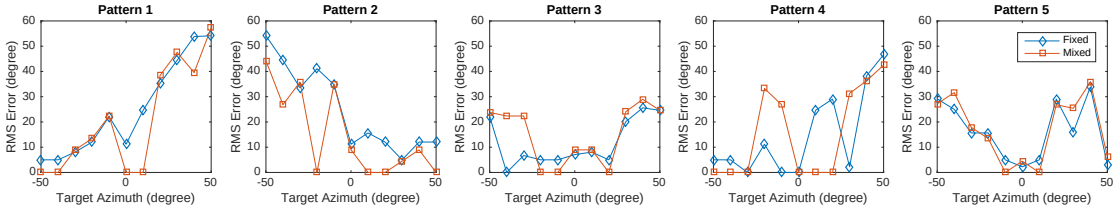


Figure 5.6: RMS errors as a function of the target location with various masking patterns.

Figure 5.6 shows RMS errors as a function of the target location for various masking patterns. In the 'fixed' conditions, the locations of the masking speakers are cued, while in the 'mixed' conditions the masker locations are not cued.

6 Summary and discussion

6.1 Summary

The report has described recent progress on feature selection and semantic labelling in the Two!Ears system. Using a classifier-based approach, Chapter 3 investigated the effectiveness of different features generated by the AFE for the classification of acoustic sources. Four classes of target sounds drawn from the NIGENS database were used for evaluation, and used to construct evaluation tasks of increasing complexity: (i) target sounds emanating from different spatial directions (ii) target sounds corrupted by white noise (iii) mixtures of target and interfering sounds, in which the sources may originate from the same or different spatial locations. It was found that an approach in which the Lasso technique is initially used, followed by construction of a linear classifier using support vector machine learning on the selected features, gave high performance while also drastically reducing the number of features to be computed. If feature selection is used in this way, a classifier constructed with multiconditional training can generalize well to other conditions not present in the training set, while at the same time being computationally efficient. Applying feature selection to construct specialized classifiers for a specific task and condition we find an even better generalization performance to test data from the same condition. This comes at the expense of a reduced performance when the classifiers are applied to data from a different condition. Specialised classifiers, however, could still be beneficial if exploited in a top-down feedback-loop, where inference on the condition is used to select the classifier.

Chapter 4 focused on learning and semantic labelling. Location and motion parameters are derived by a novel approach in which deep neural networks (DNNs) are used to map binaural features to the source azimuth. This approach consistently outperforms a previous approach in which the distributions of binaural features were modelled using Gaussian mixture models (GMMs). Furthermore, an approach has been developed for estimating the location and motion of acoustic sources that takes into account head movements, using a nonlinear dynamical system in which a control input is used to steer the head towards the desired orientation. The proposed framework is capable of tracking the position and angular velocity of a moving sound source over time. Building on the feature selection work described in Chapter 3, approaches for learning and recognising source types are described. Machine learning techniques were used to construct classifiers that have very good overall

classification performance, based on short signal blocks without any contextual information. Classification performance degrades as the signal-to-noise ratio (SNR) is reduced, but remains high for SNRs as low as 0 dB.

Chapter 4 also presented two different approaches to audio-visual integration for speech recognition; direct concatenation of audio and visual features, and joint recognition within a graphical model. Substantial performance gains are achieved using the latter approach. Evaluations were conducted using different decoders (a coupled HMM framework and turbo decoding) and two different types of observation uncertainties (uncertainty decoding and noise-adaptive linear discriminant analysis (NALDA)). The combination of turbo decoding and NALDA gave the highest performance. Other approaches for integrating graphical models in the Two!Ears system were also discussed, in two respects. First, approaches for refining segmentation results by introducing graphical-model-based techniques from the field of computer vision were described. Specifically, Markov random fields have been used to segment a spectrogram into groups of time-frequency units that belong to different sources. A proposal is also set out to use graphical models for high-level analysis of acoustic scenes, providing a flexible framework which can be used in conjunction with low-level processing components. Semantic labels generated by the approaches described above (e.g., source type classifiers and localisation modules), serve as observations for a Bayesian Network that describes specific properties of the acoustic scene; the remaining variables (e.g., the number of sources present) can then be inferred via approximate inference techniques, given the observations.

Chapter 5 presented preliminary work on the DASA-1 scenario, in which the task is to identify the location of a female voice in the presence of four concurrent male-speech maskers. Using the approaches described above, it is shown that the five concurrent voices in this scenario can be localised and segmented. An approach for gender recognition was also described, which allows the system to discriminate the male and female voices. Finally, a scheme for using top-down feedback in the system is reported, which allows the Two!Ears system to exploit information about the source types present and the locations of masker sounds.

6.2 General discussion

Work over this period has progressed well, with substantial achievements reported here in the areas of feature selection, source classification and semantic labelling, segmentation, audio-visual integration and graphical modelling. The case study discussed at the end of this report is indicative of the focus in the remainder of the project; the whole system will be evaluated on a number of well-defined scenarios in order to drive development forward and give an objective measure of the system performance. Further discussion of the scenarios to be used is found in Deliverable D6.1.2.

Acronyms

AFE auditory front-end

BN Bayesian network

CASA computational auditory scene analysis

CPT conditional probability table

DASA danymic auditory scene analysis

EM expectation-maximisation

GM graphical model

GMM Gaussian mixture model

HRIR head-related impulse response

ILD interaural level difference

ITD interaural time difference

ICC interaural cross-correlation

KEMAR Knowles Electronics manikin for acoustic research

KS knowledge source

MAP maximum *a posterioi*

MRF Markov random field

QDA quadratic discriminant analysis

PDF probabilty density function

QoE quality of experience

RV random variable

SVM support vector machine

UKF Unscented Kalman Filter

Bibliography

Algazi, V. R., Avendano, C., and Duda, R. O. (**2001**), "Elevation localization and head-related transfer function analysis at low frequencies," *J. Acoust. Soc. Am.* **109**(3), pp. 1110–1122. (Cited on page 36)

Banerjee, A., Dhillon, I. S., Ghosh, J., and Sra, S. (**2005**), "Clustering on the Unit Hypersphere Using Von Mises-Fisher Distributions," *J. Mach. Learn. Res.* **6**, pp. 1345–1382. (Cited on pages 72 and 73)

Barker, J., Vincent, E., Ma, N., Christensen, H., and Green, P. (**2013**), "The PASCAL CHiME speech separation and recognition challenge," *Computer Speech and Language* **27**(3), pp. 621–633. (Cited on pages 54 and 55)

Bengio, Y. (**2009**), "Learning deep architectures for AI," *Foundations and Trends in Machine Learning* **2**(1), pp. 1–127. (Cited on page 33)

Besbes, O., Boujemaa, N., and Belhadj, Z. (**2011**), "Embedding Gestalt Laws on Conditional Random Field for Image Segmentation," in *Proceedings of the 7th International Conference on Advances in Visual Computing - Volume Part I*, Springer-Verlag, Berlin, Heidelberg, ISVC'11, pp. 236–245. (Cited on page 58)

Best, D. and Fisher, N. (**1981**), "The BIAS of the maximum likelihood estimators of the von mises-fisher concentration parameters," *Communications in Statistics - Simulation and Computation* **10**(5), pp. 493–502. (Cited on page 73)

Bishop, C. M. (**2006**), *Pattern Recognition and Machine Learning*, Springer. (Cited on page 61)

Blauert, J. (**1997**), *Spatial hearing - The psychophysics of human sound localization*, The MIT Press, Cambride, MA, USA. (Cited on page 33)

Bregman, A. S. (**1990**), *Auditory scene analysis: The perceptual organization of sound*, The MIT Press, Cambridge, MA, USA. (Cited on page 62)

Brown, G. and Cooke, M. (**1994**), "Computational auditory scene analysis," *Comput. Speech. Lang.* **8**, pp. 297–336. (Cited on page 79)

Brungart, D. S. and Rabinowitz, W. M. (**1999**), "Auditory localization of nearby

sources. Head-related transfer functions," *J. Acoust. Soc. Am.* **106**(3), pp. 1465–1479. (Cited on page 36)

Cooke, M., Barker, J., Cunningham, S., and Shao, X. (**2006**), "An audio-visual corpus for speech perception and automatic speech recognition," *Journal of the Acoustical Society of America* **120**(5), pp. 2421–4. (Cited on page 54)

David, H. A. and Nagaraja, H. N. (**2003**), *Order Statistics*, Wiley, 3 ed. (Cited on page 10)

Dempster, A. P., Laird, N. M., and Rubin, D. B. (**1977**), "Maximum likelihood from incomplete data via the EM algorithm," *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B* **39**(1), pp. 1–38. (Cited on page 71)

Deng, L., Droppo, J., and Acero, A. (**2005**), "Dynamic compensation of HMM variances using the feature enhancement uncertainty computed from a parametric model of speech distortion," *IEEE Trans. Speech and Audio Processing* **13**(3), pp. 412–421. (Cited on page 56)

Hammersley, J. M. and Clifford, P. (**1971**), "Markov field on finite graphs and lattices," . (Cited on page 60)

Hastie, T., Tibshirani, R., and Friedman, J. (**2001**), *The Elements of Statistical Learning*, Springer Series in Statistics, Springer New York Inc., New York, NY, USA. (Cited on page 76)

Hosking, J. R. M. (**1990**), "L-moments: analysis and estimation of distributions using linear combinations of order statistics," *Journal of the Royal Statistical Society. Series B (Methodological)* **52**(1), pp. 105–124. (Cited on page 10)

Hummersone, C., Mason, R., and Brookes, T. (**2010**), "Dynamic precedence effect modeling for source separation in reverberant environments," *IEEE Transactions on Audio, Speech, and Language Processing* **18**(7), pp. 1867–1871. (Cited on page 35)

Hung, W.-L., Chang-Chien, S.-J., and Yang, M.-S. (**2012**), "Self-updating clustering algorithm for estimating the parameters in mixtures of von Mises distributions," *Journal of Applied Statistics* **39**(10), pp. 2259–2274. (Cited on page 73)

Kidd, G., Best, V., and Mason, C. R. (**2008**), "Listening to every other word: Examining the strength of linkage variables in forming streams of speech," *Journal of the Acoustical Society of America* **124**, pp. 3793–3802. (Cited on page 67)

Kolossa, D. and Haeb-Umbach, R. (Eds.) (**2011**), *Robust Speech Recognition of Uncertain or Missing Data - Theory and Applications*, Springer. (Cited on page 56)

Kolossa, D., Zeiler, S., Saeidi, R., and Fernandez Astudillo, R. (**2013**), "Noise-Adaptive

LDA: A New Approach for Speech Recognition Under Observation Uncertainty," *IEEE Signal Processing Letters* **20**(11), pp. 1018–1021. (Cited on page 57)

Kopčo, N., Best, V., and Carlisle, S. (**2010**), "Speech localization in a multitalker mixture," *Journal of the Acoustical Society of America* **127**(3), pp. 1450–1457. (Cited on pages 67, 68, and 77)

Lagrange, M., Martins, L. G., Murdoch, J., Member, S., and Tzanetakis, G. (**2007**), "Normalized Cuts for predominant melodic source separation," in *Proceedings of the International Conference on Music Information Retrieval*, pp. 163–164. (Cited on page 58)

Lakra, S., Singh, J., and Singh, A. (**2013**), "Automated pitch-based gender recognition using an adaptive neuro-fuzzy inference system," in *Intelligent Systems and Signal Processing (ISSP), 2013 International Conference on*, pp. 82–86. (Cited on page 75)

Luettin, J., Potamianos, G., and Neti, C. (**2001**), "Asynchronous Stream Modelling for Large Vocabulary Audio-Visual Speech Recognition," in *Proc. ICASSP*, pp. 169–172. (Cited on page 56)

Ma, N., Brown, G. J., and Gonzalez, J. A. (**2015**a), "Exploiting top-down source models to improve binaural localisation of multiple sources in reverberant environments," in *Proc. Interspeech'15*. (Cited on pages 78 and 79)

Ma, N., Brown, G. J., and May, T. (**2015**b), "Robust localisation of of multiple speakers exploiting deep neural networks and head movements," in *Proc. Interspeech'15*. (Cited on pages 33 and 69)

Ma, N., May, T., Wierstorf, H., and Brown, G. J. (**2015**c), "A machine-hearing system exploiting head movements for binaural sound localisation in reverberant conditions," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. (Cited on page 35)

MacQueen, J. B. (**1967**), "Some methods for classification and analysis of multivariate observations," in *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297. (Cited on page 71)

Manfredi, G., Devy, M., and Sidobre, D. (**2015**a), "Textured Object Recognition: Balancing Model Robustness and Complexity," in *16th Int. Conf. on Computer Analysis of Images and Patterns (CAIP'2015)*, La Valette, Malta. (Cited on page 53)

Manfredi, G., Devy, M., and Sidobre, D. (**2015**b), "Visual Localisation from Structureless Rigid Models," in *Int. Conf. on Advanced Concepts for Intelligent Vision Systems (ACIVS' 2015)*, Catane, Italy. (Cited on page 53)

May, T., Ma, N., and Brown, G. J. (**2015**), "Robust localisation of multiple speakers

exploiting head movements and multi-conditional training of binaural cues," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.* (Cited on page 35)

Ren, Y., Tang, H., and Wei, H. (**2011**), "A Markov Random Field Model for Image Segmentation Based on Gestalt Laws," in *Neural Information Processing*, edited by B.-L. Lu, L. Zhang, and J. Kwok, Springer Berlin Heidelberg, vol. 7064 of *Lecture Notes in Computer Science*, pp. 582–591. (Cited on pages 58 and 60)

Scheler, D., Walz, S., and Fingscheidt, T. (**2012**), "On iterative exchange of soft state Information in two-channel automatic Speech Recognition," in *Proc. ITG Facht. Sprachkomm.* (Cited on page 56)

Schölkopf, B., Smola, A. J., Williamson, R. C., and Bartlett, P. L. (**2000**), "New Support Vector Algorithms," *Neural Comput.* **12**(5), pp. 1207–1245. (Cited on page 71)

Schymura, C., Winter, F., Kolossa, D., and Spors, S. (**2015**), "Binaural Sound Source Localisation and Tracking using a dynamic Spherical Head Model," in *Interspeech*, Dresden, Germany. (Cited on pages 35 and 36)

Varga, A. and Moore, R. (**1990**), "Hidden Markov model decomposition of speech and noise," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pp. 845–848. (Cited on page 79)

Wang, D. L. and Brown, G. J. (Eds.) (**2006**), *Computational Auditory Scene Analysis: Principles, Algorithms and Applications*, Wiley/IEEE Press. (Cited on pages 69 and 78)

Wierstorf, H., Geier, M., Raake, A., and Spors, S. (**2011**), "A Free Database of Head-Related Impulse Response Measurements in the Horizontal Plane with Multiple Distances," in *130th Convention of the Audio Engineering Society.* (Cited on page 71)

Zeiler, S., Nickel, R., Ma, N., Brown, G., and Kolossa, D. (**2016**), "Robust audiovisual speech Recognition Using Noise-Adaptive Linear Discriminant Analysis," in *submitted to Proc. ICASSP.* (Cited on page 56)