Combining Multiple Classifiers for Age Classification

Charl van Heerden, Etienne Barnard

Human Language Technologies Research Group
Meraka Institute, CSIR, South Africa
cvheerden@csir.co.za, ebarnard@csir.co.za

Abstract

We compare several different classifier combination methods on a single task, namely speaker age classification. This task is well suited to combination strategies, since significantly different feature classes are employed. Support vector machines (SVMs) are trained on two different types of feature classes to estimate posterior class probabilities. The posteriors from these classifiers are combined using different combination rules and functions described in the literature. A novel age classifier is also developed by using an SVM to predict posterior class probabilities using two different types of classifier outputs; gender classification results and regression age estimates. We show that for combining posterior probabilities, simple combination rules such as the product rule perform surprisingly well as opposed to trainable combination strategies that require a significant amount of data and training effort.

1. Introduction

There are ample evidence and examples in the literature which show that classifier combination or fusion can improve on the accuracy of any one of the individual constituting classifiers [1, 2, 3, 4, 5, 6, 7].

The reason for using different classifiers (and hence requiring classifier combination strategies), has changed over the last couple of years. In the past, classifier combination was necessary, mainly to make the classification process more efficient by using multiple stages of classification. Initial stages could separate difficult from easier classes, with only the difficult classes requiring more expensive feature sets or classification algorithms. On the other hand, successive stages could be used to gradually reduce the number of possible classes [2], in much the same way as two class classifiers are used to enable multiclass classification in support vector machines [8].

Today however, the motivation for multiple classifier combination is mainly the quest for optimal accuracy:

- For many problems, different feature classes exist for which one may wish to train and optimize different classifiers (with each individual classifier best suited to its particular task). A good example of this is short term, frame-based features as opposed to long-term, higher level features for speaker verification [9]. It has been observed that classifier combination is particularly effective when different feature classes are employed [2]
- There are many different classification algorithms available such as Gaussian mixture models (GMMs), support vector machines (SVMs) and neural networks (NN), as well as different ways in which to use them, for example K-nearest neighbor (KNN) with different numbers of neighbors [1, 6, 2]. These algorithms tend to err in subtly different ways, thus creating an opportunity for improved performance from their combination.

Several different combination strategies exist; these can be classified based on a set of criteria set out in [7]. The main distinction is whether or not a combination strategy needs training data to estimate parameters for some combination function.

Our goal in this paper is twofold: 1) We want to compare many of these combination strategies on a single problem and 2) we aim to gain insight on how well trainable as opposed to non-trainable methods perform.

The particular problem that we investigate in order to compare all of these combination strategies, is that of speaker age classification. This problem is well suited to the classifier combination strategy as outlined in the motivation above, as significantly different feature classes are employed. Also, it is a hard problem (with Bayes errors around 50%, as discussed below), and no single classifier performs very well on this task.

The rest of this paper is organized as follows: in section 2, we give an overview of the different types of strategies for classifier combination and also of the background and aim of age classification. The different feature classes used are discussed in section 3, while the design of the classifiers used in this study are discussed in section 4. Different classifier combination strategies and their application to the age classification task are discussed in section 5, with the corresponding results, discussion and conclusion following in sections 6, 7 and 8 respectively.

2. Background

A general overview of classifier combination is given in section 2.1. Some background on the age classification task is given in section 2.2, while the corpus that was used for the experiments reported in this paper is described in section 2.3.

2.1. Classifier combination overview

Combination strategies can be grouped into different categories, based on specific criteria [7]. One such distinction that is the focus of the current paper, is between "combination rules" and "combination functions", the first indicating simple rules such as the sum or product rule, while the second includes more complicated functional combinations that require training, such as support vector machines. Combination strategies used in this paper will also operate on the "score level" as opposed to the "feature level". Furthermore, only "non-ensemble combinations" as opposed to "classifier ensembles" such as bagging and boosting, will be considered, since we are focusing on a small, fixed set of classifiers.

Another set of criteria for distinguishing combination strategies is based on the expected output from classifiers, and can be divided into 3 main categories [5]:

- Single class labels. The classifier assigns a test vector to one of N classes and provides only the assigned class label as output
- Ranked class labels. The classifier outputs class labels ranked in the order of likelihood of the test vector originating from a particular class
- Real valued outputs. The classifier outputs a real value that somehow denotes the likelihood of the vector belonging to a particular class. Posterior probabilities are often used as real valued outputs.

We investigate combination strategies based on all 3 of the above mentioned categories.

2.2. Age classification overview

Age and gender classification from speech has been a topic of interest from as early as the 1950's [10]. More recently, workshops have been organized to compare existing approaches to age and gender classification on a common database (German SpeechDat II corpus) [11] and the age classification task was also formalized as the classification of a speaker according to seven age/gender groups. Approaches that have been employed successfully include classification based on phone recognition and direct age classification. For the latter case, two main classes of features have been most popular: long-term (mostly prosodic) features and short-term features based on Mel frequency cepstral coefficients (MFCCs). Extensive work has been done on refining and measuring the significance of the long term features [12], as well as on ways to optimally combine the two feature classes [13].

Regression to estimate speaker ages has recently been suggested as an alternative to age-category classification [14]. Since the focus in [14] was to compare different feature types, the relative performance of regression-based and classification-based approaches was not investigated. We perform such a comparison (using support vector machine regression), and also show how regression can be combined with gender classification to perform the standardized 7-class task mentioned above. In fact, this regression-based approach is somewhat more accurate than 7-class classifiers trained on either of the above-mentioned feature classes.

2.3. Corpus and classification task

The corpus that was used for the age classification study consisted of speech from approximately 700 German speakers, recorded at 8000 Hz. There were 18 utterances per session, with up to 6 sessions per speaker. Utterances were between 1 and 6 seconds in duration (the distribution of durations is shown in figure 1), with the total corpus size amounting to 47 hours.

Approximately 90% of the speakers were labeled and subsequently used for the experiment. This set was divided into 3 sets: a training set (40%), a development set (30%) and a test set (30%), with no speakers being in more than one set.

The age classification task is similar to one formalized at a workshop organized by Deutsche Telekom [11]. Given a single utterance, a system needs to classify the speaker as coming from one of 7 age/gender groups. These groups are 1) children (< 13 years), 2) young female or 3) male ($13 - 19$), 4) adult female or 5) male ($20 - 64$) and 6) senior female or 7) male (> 64).

3. Feature classes

Two classes of features were used to perform the age classification task: long-term (mostly prosodic) features (LTF) and short-term, supervector features (SPV) derived from MAP adaptation of Gaussian mixture models (GMMs).

Long-term features consisted of 22 features, comprising pitch, jitter, shimmer and intensity. These particular features were chosen since they are known to correlate with speaker age and gender. For example, a high average micro-variation in voice frequency (jitter) may be due to an age-related deterioration of the glottis. The specific values used included the mean, minimum, maximum, standard deviation, and deltas of each of the features. These were extracted using Praat[15] employing a cross-correlation method for pitch period analysis with a stepsize of $10ms$. A detailed description of all features is provided in [16].

In addition to these features, formants were also extracted from the voiced sections of the spoken audio. Praat was used to extract the first four formants using Burg's algorithm. A sliding window with a length of $25ms$ and a stepsize of $20ms$ was used. The maximum formant frequency was specified to be 5500 Hz, which is a common choice for adult females. One would typically choose a lower value for adult males and a much higher value for young children if the gender was known beforehand. The mean and standard deviation was then calculated for each formant, as well as its first derivative. The pitch corresponding to the period over which the formants were extracted was also added to create a 20-dimensional vector.

As **short-term features**, mel frequency cepstral coefficients (MFCCs) were extracted from all utterances using the HTK toolkit with a stepsize of $5ms$, a window length of $30ms$, and a dimension of 12. A 128-mixture Gaussian mixture model (GMM) was then trained to model the coefficients, with MAP adaptation applied to update the means and weights for all mixtures, given a new utterance. The resulting adapted means were then concatenated to form a $1,536$ dimensional supervector (12 coefficients $*$ 128 mixtures).

4. Classification design

In order to perform regression, the utterance vectors of both feature classes were annotated with the true ages of the speakers, as provided by the speakers during data collection. A support vector machine was then trained, with the objective of finding a function f that predicts the target ages with at most ϵ years deviation, while being as flat as possible [17]. Given these models, regression was then performed by mapping the test vectors into a high dimensional feature space, computing dot products with the transformed training vectors and adding the results using precomputed weights to obtain the final age estimate. Regressors were trained for both feature classes with LIBSVM, using the radial basis kernel function (RBF) [18]. It was found that the performance of these regressors, as well as of the classifiers described below, depends strongly on the parameters employed during training. For the regressors, these parameters are ϵ (the regression error that is allowed before a particular sample is penalized), C (which controls the trade-off between margin width and training-set error) and γ (the width of the RBF kernel), whereas classification involves C and γ only. Both regressors and classifiers were optimized in terms of ϵ, C and γ using 10-fold cross validation and grid searches on the training set. All folds contained data from distinct speakers and were balanced based on the number of speakers per fold.

Since the task of age classification in the commonly-used format requires a distinction between for example young males and young females, and since the regression estimate is insensitive to the gender of a speaker, it was necessary to train classifiers for distinguishing children, males and females from each other. Two gender classifiers were thus trained to estimate the posterior probabilities of an utterance originating from children, males or females, using the LTF and SPV feature classes respectively.

A second level of classification was necessary to combine the outputs from the gender classification and the regressors. The posteriors from the two gender classifiers were multiplied and together with the two regression outputs, a 5-dimensional vector was created. These vectors were then used to train a 7-class classifier.

Since the combination described above entails using the output of classification and regression results, we had to "generate" training data. This was accomplished by dividing the training set into 10 folds (having distinct speakers) and then following a round-robin approach to train gender classifiers and regressors on 90% and repeatedly classifying the remaining 10%. All classifiers used in this round-robin approach used the same parameters for a particular feature set. Another grid search was then used to optimize the 7-class classifier using cross validation.

In order to benchmark our regression-based classifier against existing techniques, we trained 7-class classifiers on both feature classes. The SVMs were trained using an RBF kernel, and grid searches combined with 10-fold cross validation were again employed to search for the optimal values of C and γ for each of the classifiers.

5. Classifier combination strategies

Multiple classifier combination strategies were employed; the sum, max, min, majority voting, median and product rules, as well as using an SVM for combination, or weighting the results based on [4]. Another strategy implemented was a variant of majority voting, the Borda Count [19]. Behavior Knowledge Space (BKS), first introduced in [20], was also used, both as an individual classifier combination strategy, as well as an additional classifier that could provide posterior probabilities. A description of all but the last four combination strategies, as well as the corresponding advantages and disadvantages are well covered in [2].

5.1. Majority voting rule & Borda Count

The majority voting rule was slightly adapted from [2] in order to handle cases where no consensus could be reached (in this case, a 3-way tie). A simple strategy was used where "experts" were allowed to change their vote based on a cost function. Only one expert was allowed to change their vote at any one time, with the expert allowed to change their vote being selected based on the least accumulated cost to change their vote. Cost was calculated as the difference between the currently selected posterior probability and their next highest posterior probability. In this fashion, an expert not sure about a particular choice will be allowed to change their vote, while an expert that was very confident in his decision would be much less likely to change their vote.

The same concept was followed for the case where there were ties when using Borda Count. The only difference was that instead of allowing experts to change their votes, the confidence in their initial vote was reduced by half.

5.2. Behavior Knowledge Space

The BKS method was actually designed to combine outputs on the "single class label" level. In order to adhere to this requirement, every set of posterior probabilities corresponding to a single test vector, was mapped to the class label corresponding to the maximum posterior probability. This was done for each set of classifier results independently. Posteriors were then generated for each combination of the k classifier outputs ($x_1 ... x_k$) by considering the frequency C of the associated combination vector indicating the current class label, ω_i [20]:

$$\hat{P}(\omega_i | x_1, ..., x_k) = \frac{C(\omega_i | x_1, ..., x_k)}{\sum_i C(\omega_i | x_1, ..., x_k)} \qquad (1)$$

5.3. Support vector machine combination classifier

An SVM was trained to combine the posteriors of the 3 7-class classifiers. An RBF kernel was used, with C and g optimized using grid searches and cross validation.

6. Results

The accuracies achieved by the three basic classifiers (7-class classifiers based on the two types of features, as well as the regression-based classifier) are indicated as a function of the utterance duration in Figure 1. That figure also shows the accuracy for the "product" combination strategy. The overall results (that is, with all durations combined) from the different combination strategies described in section 5, are displayed in order of descending accuracy in table 1.

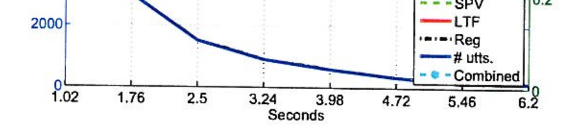


Figure 1: *Accuracy of the different classifiers vs utterance duration, along with the distribution of utterance durations in the corpus employed.*

6.1. Majority voting rule analysis

The results from the majority voting rule are shown in table 2 and should be investigated for future classifier combination improvements. One may want to apply different combination strategies based on the individual classifier results.

Table 1: *Different classifier combination results. The first 3 results*[1,2,3] *show the individual classifier performance on both the 7-class problem (columns two and three), as well as the gender classification task (last column). The remaining results show different classifier combination strategies for combining all three systems on the 7-class problem (column two), combining systems* 1 *and* 2 *on the 7-class task (column three) and combining systems* 1 *and* 2 *on the gender task (last column).*

Combination Strategy	Accuracy		
	cl7*3	cl7*2	cl3
LTF[1]	45.67	45.67	86.38
SPV[2]	45.26	45.26	83.93
GID/regression[3]	48.38	48.38	-
Weighted [4]	50.75	49.42	88.06
Product	50.73	49.38	88.51
Sum/Average	50.62	48.84	88.47
BKS mult	50.54	49.53	88.66
Median	50.44	48.84	88.47
Borda count	50.39	48.37	88.19
Majority vote	50.29	-	-
BKS sum	50.22	49.10	88.53
Min	49.98	49.33	88.46
SVM	49.42	49.39	87.15
Max	49.06	48.18	88.39
BKS	48.08	43.86	86.38

Table 2: *Majority voting rule results, showing the number of test vectors receiving a particular amount of votes, with the corresponding accuracies of the set of vectors receiving at least x votes.*

# of votes	# of test vectors	Accuracy
1+	12735	49.05
2+	11133	52.49
3	4275	65.92

6.2. GID posteriors & regressions estimates combination

The combining classifier trained to combine GID posteriors and age estimates from regression was described in section 4. The results from this classifier, as well as those of the individual constituent classifiers, are shown in table 3. A confusion matrix summarizing the performance of the combined GID systems is also shown in table 4.

To further analyze this approach, we estimated the posterior probability of each of the 7 classes, given the regressor and gender ID outputs. This is graphically depicted in figure 2. Looking at figure 2 (c), one can for example see that when the gender ID predicted that the speaker is male and the regressor estimate is 66, there is an approximately 50% probability of the speaker actually being and old male.

7. Discussion

The results from the different classifier combination strategies on the posteriors will be discussed in section 7.1, while the results from the SVM combination of the posteriors and regression estimates will be discussed in section 7.2.

Table 3: *Accuracies and parameters of all classifiers, where the results are MSE and SCC respectively for the regressors. Systems B and D are results reported on the same feature classes for a similar task [11]*

	C	g	eps	Acc
LTF reg	31.623	0.316	10	363.3/0.226
SPV reg	5.580	0.032	8	338.6/0.251
LTF cl7	3162.278	0.003		45.67
SPV cl7	5.667	0.01		45.26
LTF cl3	3.162	0.316		86.38
SPV cl3	3.162	0.01		83.93
Reg+GID	1	10		48.38
3 class combined				88.51
7 class combined				50.73
System B (LTF)				40
System D (SPV)				42

Table 4: 3-*class confusion matrix for the combined (product rule) LTF and SPV feature classes*

	C	F	M
C	832	705	171
F	184	5244	257
M	11	135	5196

7.1. Combining similar classifier outputs

The results in table 1 show that simple combination rules perform surprisingly well compared to combination functions that require parameters to be estimated in a separate training process. In particular, the product rule performs better than any of the simple combination rules and better than all but 1 combination function when combining the three 7-class classifiers. The weighted sum, with weights calculated on training data, according to the algorithm proposed in [4], performs marginally better than the product rule on that task. The BKS combination strategy on the other hand, performs significantly worse than the other combination strategies on the 7-class tasks. This can be attributed to the fact that almost all the other combination strategies effectively utilize the additional information inherent in the posterior probabilities available from each of the classifiers, while BKS requires hard decisions to start off with. The posteriors generated from the BKS combination strategy seem to be very promising though. These additional posteriors were combined with the existing classifier outputs using the sum and product rules, with the results shown in table 1. For both the 7-class classifier combination problems, these posteriors, combined with the existing classifier posteriors using the product rule, perform better than any other approach. The fact that the 3 7-class classifier combination approach does not seem to benefit, can probably be attributed to the fact that BKS requires a significant amount of training data to reliably estimate the posteriors in this particular case. This is evident from the number of posteriors P_N that have to be estimated:

$$P_N = K^{C+1} \tag{2}$$

where K is the number of classes and C is the number of classifiers being combined. From equation 2, one can see that relatively few posteriors need to be estimated for the two 7-class and 3-class combination problems (343 and 27 posteriors respectively). For the 3 7-class combination problem, 2401 posteriors need to be estimated.

7.2. Gender ID and regression estimate combination

The results from the svm combined gid and regression estimates confirm that one can effectively fuse different types of classifier results. This combined classifier performs better than any of the two direct 7-class classifiers. Figure 2 gives some insight as to how this classifier performs and which classes are easily confused. Looking at figure 2 (b) for example, one can clearly see that children and young women will often be confused when the regressor estimate is less than 24 years. This is also reflected in table 4, where it is clear that children and women are the two most confusable classes. We also see that senior males and females are accurately indicated for regression estimates above about 56 years, and that middle-aged men are particularly hard to classify.

8. Conclusion

This paper compared different classifier combination strategies for the age classification task. This task is well suited to classifier combination because of the significantly different feature classes that were employed. The feature classes used were long term prosodic features and short term (frame-based) supervector features. Support vector machines were trained to obtain both 7-class and 3-class (gender) classification results from each feature class respectively, in the form of posterior probabilities. SVM regressors were also trained to obtain age estimates for each test vector from each of the feature classes. A novel 7-class classifier was then created by combining two different types of real valued classifier outputs; GID posterior probabilities and age estimates from the regressors. A combination classifier (SVM) was trained for this purpose, giving real valued decisions in the form of 7-class posterior probabilities.

Several combination strategies which are well covered in literature [5, 2] were then implemented to combine the 3 7-class classifier outputs: the product rule, the sum/average rule, the median rule, the max rule, the min rule, the majority voting rule, a simple weighting rule [4], Borda count, BKS and the use of a combination classifier (SVM). The results showed that combination functions that don't require any training, such as the product rule, performs almost just as well, if not better than many of the trainable combination functions. It was also interesting to observe that even though SVMs perform exceedingly well on the individual feature classes, they perform significantly worse on the posterior combination task when compared to simple combination rules such as the sum or product rules..

9. Acknowledgements

We thank Dr. Christian Müller and Michael Feld from DFKI for providing extracted features on the dataset reported on in this paper. We also thank Dr. Marelie Davel for many discussions related to this work.

10. References

[1] L. Xu, A Kryzak, and C.V. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE transactions on Systems, Man and Cybernetics*, vol. 22, no. 3, pp. 418–435, June 1992.

[2] J. Kittler, "Combining classifiers: A theoretical framework," *Pattern Analysis & Applications*, vol. 1, no. 1, pp. 18–27, March 1998.

[3] F.M. Alkoot and J. Kittler, "Experimental evaluation of expert fusion strategies," *Pattern Recognition Letters*, vol. 20, no. 11-13, pp. 1361–1369, November 1999.

[4] Oriol Ramos Terrades, Ernest Valveny, and Salvatore Tabbone, "Optimal classifier fusion in a non-bayesian probabilistic framework," *IEEE Transactions on Pattern Analysis abd machine intelligence*, vol. 31, no. 9, pp. 1630–1644, September 2009.

[5] David M.J. Tax, Martijn van Breukelen, Robert P.W. Duin, and Josef Kittler, "Combining multiple classifiers by averaging or by multiplying?," *Pattern Recognition*, vol. 33, no. 9, pp. 1475–1485, September 2000.

[6] Robert R.W. Duin, "The combining classifier: to train or not to train?," in *International Conference on Pattern Recognition*, Quebec City, Canada, August 2002, pp. 765–770.

[7] Sergey Tulyakov, Stefan Jaeger, Venu Govindaraju, and David Doermann, "Review of Classifier Combination Methods," in *Studies in Computational Intelligence: Machine Learning in Document Analysis and Recognition*, Hiromichi Fujisawa Simone Marinai, Ed., pp. 361–386. Springer, 2008.

[8] Chih-Wei Hsu and Chih-Jen Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, March 2002.

[9] Douglas Reynolds, Walter Andrews, Joseph Campbell, Jiri Navratil, Barbara Peskin, Andre Adami, Qin Jin, David Klusacek, Joy Abramson, Radu Mihaescu, Jack Godfrey, Doug Jones, and Bing Xiang, "The supersid project: Exploiting high-level information for high-accuracy speaker recognition," in *ICASSP*, Hong Kong, April 2003, pp. 784–787.

[10] Edward D. Mysak, "Pitch duration characteristics of older males," *Journal of Speech and Hearing Research*, vol. 2, pp. 46–54, 1959.

[11] Florian Metze, Jitendra Ajmera, Roman Englert, Udo Bub, Felix Burkhardt, Joachim Stegmann, Christian Müller, Richard Huber, Bernt Andrassy, Josef G. Bauer, and Bernhard Littel, "Comparison of four approaches to age and gender recognition for telephone applications," in *ICASSP*, Honolulu, Hawaii, April 2007, pp. 1089–1092.

[12] C. Müller, "Automatic recognition of speakers' age and gender on the basis of empirical studies," in *Interspeech*, Pittsburgh, Pennsylvania, September 2006.

[13] C. Müller and F. Burkhardt, "Combining short-term cepstral and long-term prosodic features for automatic recognition of speaker age," in *Interspeech*, Antwerp, Belgium, August 2007, pp. 2277 – 2280.

[14] Werner Spiegl, Georg Stemmer, Eva Lasarcyk, Varada Kolhatkar, Andrew Cassidy, Blaise Potard, Stephen Shum, Young Chol Song, Puyang Xu, Peter Beyerlein, James Harnsberger, and Elmar Nth, "Analyzing features for automatic age estimation on cross-sectional data," in *Interspeech*, Brighton, England, Sept 2009, pp. 2923 – 2926.

[15] P. Boersma, *Praat, a system for doing phonetics by computer*, Amsterdam: Glott International, 2001.

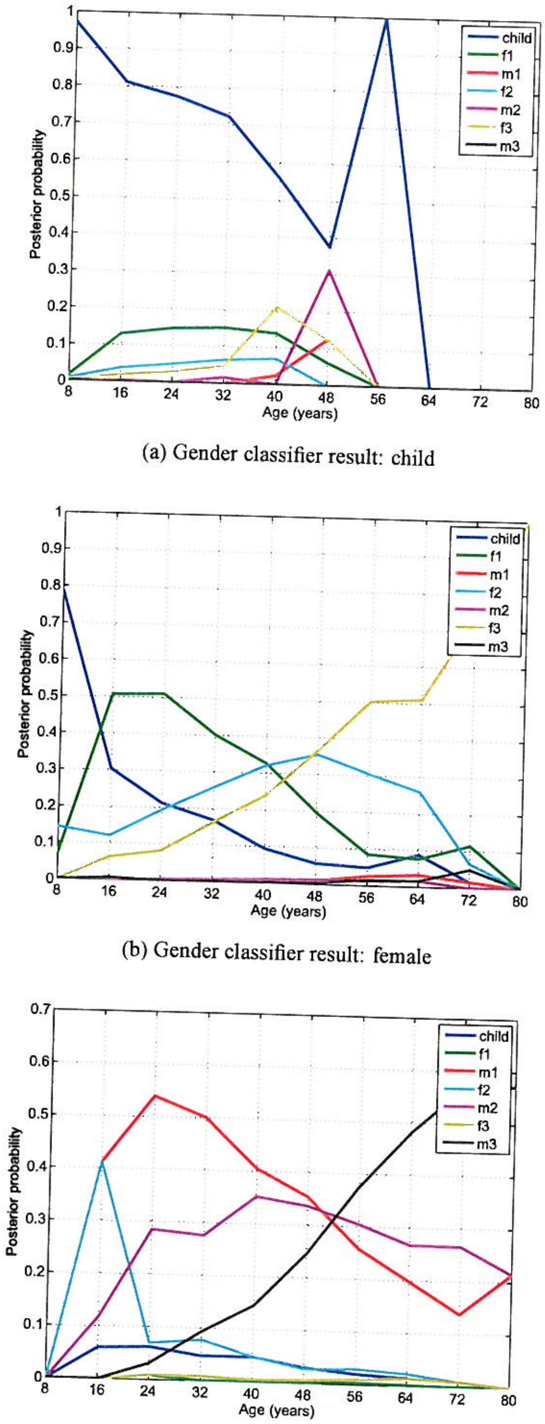
[16] Christian Müller, *Zweistufige kontextsensitive Sprecherklassifikation am Beispiel von Alter und Geschlecht*, Ph.D. thesis, Computer Science Institute, University of the Saarland, Germany, 2005.

[17] Alex J. Smola and Bernhard Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 1573–1375, August 2004.

[18] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: a library for support vector machines*, 2001, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[19] M.V. Erp, L.G. Vuurpijl, and L. Schomaker, "An overview and comparison of voting methods for pattern recognition," in *Proc. of Workshop on Frontiers in Handwriting Recognition*, Niagara on the Lake, Canada, August 2002, pp. 195–200.

[20] Y. Huang and C. Suen, "A method of combining multiple experts for recognition of unconstrained handwritten numerals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 90–94, January 1995.



(a) Gender classifier result: child

(b) Gender classifier result: female

(c) Gender classifier result: male

Figure 2: Posterior probability of each class given the gender classifier result together with the LTF regressor age estimate.