Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Feature Space SVM Adaptation for Speaker Adapted Word Prominence Detection

Andrea Schnall, Martin Ernst Heckmann

2018

Preprint:

This is an accepted article published in Computer Speech and Language. The final authenticated version is available online at: https://doi.org/ https://doi.org/10.1016/j.csl.2018.06.001

Feature-Space SVM Adaptation for Speaker Adapted Word Prominence Detection

Andrea Schnall[a], Martin Heckmann[b]

[a] *Control Methods and Robotics Lab, TU Darmstadt, Landgraf-Georg Str. 4, 64283 Darmstadt, Germany*
[b] *Honda Research Institute Europe GmbH,Carl-Legien-Str. 30, 63073 Offenbach/Main, Germany*

Abstract

Prosodic cues such as the word prominence play a fundamental role in human communication, e.g., to express important information. Since different speakers use a wide variety of features to express prominence, there is a large difference in performance between speaker dependently and speaker independently trained models. To cope with these variations without training a new speaker dependent model, in speech recognition speaker adaptation techniques such as feature-space Maximum Likelihood Linear Regression (fMLLR) turned out to be very useful. These methods are developed for GMM-HMM based classifiers under the assumption that the data can be well modeled via the mixture of a few Gaussian distributions. However, in many cases these assumptions are too restrictive. In particular a discriminative classifier such as an SVM often yields far superior results to a GMM. Therefore, we propose a new adaptation method, which adapts the data to the radial basis function kernel of the SVM. To avoid overfitting we apply two regularization terms. The first is based on fMLLR and the second is an L_1 regularization to enforce a sparse transformation matrix. We analyze the method in the context of speaker adaptation for word prominence detection, with varying amounts of adaptation data and different weights of the regularization terms. We show that our novel method clearly outperforms fMLLR-GMM and fMLLR-SVM based adaptation.

Keywords: prosody, speaker adaptation, fMLLR, SVM, prominence
2010 MSC: 00-01, 99-00
`https://doi.org/10.1016/j.csl.2018.06.001`

©< 2018 >. This manuscript version is made available under the CC-BY-NC-ND 4.0 license http://creativecommons.org/licenses/by-nc-nd/4.0/

Email addresses: `aschnall@rmr.tu-darmstadt.de` (Andrea Schnall), `martin.heckmann@honda-ri.de` (Martin Heckmann)

1. Introduction

The information in speech is not only contained in the lexical meaning of the spoken words or even in the syntax or sentence structure. Significant amounts of essential, meaning-determining information is conveyed via prosody, the way we say words. Prosody consists of rhythm, stress and intonation. This is not part of the vocabulary or the grammar, which are typically captured within speech recognition. In human-human communication, prosody can help to structure the utterance in a phrase, emphasize the novel information, differentiate between questions and statements but also show the emotion of the speaker, sarcasm and irony. Humans are also able to understand if there are different possible meanings of an utterance indicated by prosodical changes [1]. Consequently, prosody can convey a lot of valuable information for interactive systems, but still most of these systems do not include any prosodic cues [2, 3]. One reason might be that it is not easy to build a powerful detection system, since there is a large variation between different speakers. Yet without considering prosodical information, speech-processing systems stand no chance to reliably process complex human interactions [4].

In light of these considerations, prosodical information should be used for human-machine interaction systems as well, in order to improve their performance and make their use more intuitive. One prosodic cue humans use is the prominence of a word, which we define here as a greater perceived strength of one word in a sentence [5]. The determination of word prominence plays an important role in human communication. By strongly increasing the prominence of a word, for example, a correction can be indicated [1]. In state-of-the-art speech interaction systems, this way of over pronunciation often leads to even more miss-recognitions [6, 7]. Speakers also use prosodic cues to mark important information such that the important words stand out from their environment. Different approaches to extract and represent this information have been proposed. For the related and more investigated pitch accent detection often non acoustic features are used [8], i.e., syntactic, phonological, or acoustic features in combination with text-based features [9]. Yet in our scenario we want to only focus on the acoustic variations underlying word prominence. Concerning different learning approaches, in most approaches supervised learning is applied [10, 11], but some are based on semi-supervised learning [12], and unsupervised learning methods [13, 14, 15]. While a speaker dependently trained classifier for detecting word prominence can already reach quite good results, the performance drops if a speaker independent system is used. For real applications it is difficult to use a speaker dependently trained model due to the high amount of labeled data which is needed to train such a model. To overcome this problem, we use a speaker independent system and adapt it with a small amount of speaker dependent data to obtain an accuracy comparable to that of a model trained on a specific speaker. This small amount of labeled data could be collected prior to the first use of a system equipped which our algorithm or the

system could perform an unsupervised adaptation by performing the adaptation based on data collected during the interaction with high classication condence. Our experiments show that support vector machines (SVM) are a better choice for the classification than e.g. a Gaussian mixture model based classification or, in case of a low amount of training data, Deep Neural Networks (DNNs) [16]. Therefore, we investigate speaker adaptation methods for SVM-based classifiers. A standard adaptation method in speech processing, the feature-space maximum likelihood linear regression (fMLLR) [17], showed not to be successful in combination with an SVM, since the assumption that the data is well described as the superposition of a small number of Gaussian kernels in a Gaussian mixture model is questionable. Another issue might be that fMLLR does not incorporate discriminative information. In this paper we propose a method, based on the radial basis function parameters of the trained SVM model. In the adaptation these parameters are used to incorporate the information of the decision boundary. Subsequently, to avoid overfitting and reach better results with a low amount of adaptation data, we use a regularization to constrain the model. More precisely, we use an fMLLR cost term to regularize the SVM adaptation as well as an L_1 regularization. Preliminary results of these methods have been presented in [18, 19]. In this paper we substantially extend the experiments and perform a more in depth analysis of the results.

The next section will present the state of the art methods. Afterward, in section 3, we give an overview of the fMLLR (3.1) followed by the introduction of our new adaptation methods, the feature-space SVM adaptation (3.2) and the Gaussian-regularized SVM adaptation (3.3). Section 4 presents the database used followed by a description of the features in section 5. Following a short overview on the experimental setup in section 6 is the presentation of the results in section 7. We will end with a discussion of our findings in section 8 and a subsequent conclusion in our final section.

2. State of the art

Speaker adaptation techniques try in general to reach speaker dependent classification performance with only a small amount of speaker specific data. The methods can be divided in speaker normalization such as Vocal Tract Length Normalization [20], adaptation in speaker space by defining speaker 'types', model-based adaptation and feature-space adaptation. Often used methods are the model-based maximum likelihood linear regression (MLLR) [17] and maximum a posteriori (MAP) adaptation [21]. MLLR updates the parameters of a Hidden Markov Model by using statistics from labeled adaptation data to calculate a linear regression-based transformation. MAP adaptation combines the information provided by the adaptation data with some prior knowledge about the model parameters described by a prior distribution. In our case we want a speaker adaptation for the problem of word prominence detection. That means that we have, compared to the traditional speaker adaptation, no sequence recognition but a detection problem and not necessarily a Gaussian model.

In [22] a system is proposed which adapts to a driver's voice for emotion recognition in the automotive domain. The adaptation is done in the acoustic space by mean and standard deviation normalization. In [23] a combination of MAP and Gaussian mixture models (GMM) for an unsupervised adaptation to a new unlabeled data set is used. To our knowledge not much work has been done on prosodic adaptation since that and there is no other work which uses speaker adaptation methods like fMLLR for the classification of prosodic events and especially not in conjunction with support vector machines. There are some methods regarding adaptive or incremental SVMs for other problem cases like the more general domain adaptation or non-stationary environment problem. In [24] the target classifier is adapted from the existing, auxiliary classifier via a perturbation function with labeled data from the target domain. This method achieves comparable performance to an SVM trained on source and target data but has a better run time. A dynamic kernel adaptation is presented in [25] where the kernel parameter γ is automatically learned and approximate procedures to find the best model are proposed.

In [26] a recursive algorithm is proposed for SVMs which allows exact 'adiabatic' incremental learning. It does this by retaining the Kuhn-Tucker conditions on all previously seen data while adding a new point to the solution. It is extended to a general framework in [27] and for large classification problems in [28]. The procedure can be reverted by decremental unlearning of the support vectors. The work of [26] is described by [29] as the only exact incremental version of the SVM.

Daumé [30] proposed a feature replication method to augment features for cross-domain learning. The augmented features are then used to construct a kernel function for Support Vector Machine (SVM) training. The method of [31] iteratively labels unlabeled target patterns progressively and simultaneously removes some auxiliary labeled patterns. Cross-domain SVM (CD-SVM) proposed by [32] defines a weight for each source training sample based on the distance to the target domain, and retrains the SVM classifier with reweighted patterns. The domain transfer SVM (DT-SVM) proposed by Duan et al. [33] uses multiple-kernel learning to minimize the difference between the means of the source and target feature distributions. But to our best knowledge there is no method that is well suited for speaker adaptation with an SVM. Also those methods, compared to ours, mostly only allow small changes or extensions to the existing model and are predominantly for incremental learning tasks. They do not allow to rotate and translate the complete feature-space to obtain a better fit of the data. The latter is useful in the case of speaker adaptation, since in this case we do not expect a feature drift but a changed feature space. This kind of procedure, e.g., used in fMLLR, is a common approach for speaker adaptation.

3. Adaptation methods

As basis and reference for our method the fMLLR is used which will be shortly presented, followed by our proposed method, the fSVMA, and its exten-

sions with a Gaussian regularization and a sparseness term.

3.1. Feature-space MLLR (fMLLR)

The feature-space Maximum likelihood linear regression [17] is a common method in speech recognition. Compared to MLLR, which transforms mean and variance of an HMM-model, it transforms the features directly and is hence independent of the classifier and can be used in combination with, e.g., SVMs. For the calculation of the adaptation matrix we use the row-by-row updates for adaptation matrices [34]. The transformation matrix $\mathbf{W}$ consists of an affine matrix $\mathbf{A}$ and a bias vector $\mathbf{b}$:

$$\hat{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b} = \mathbf{W}\xi, \tag{1}$$

where $\xi = [1 \;\; \mathbf{x}^T]^T$ is the input vector extended with an extra element equal to unity.

To calculate $\mathbf{W}$, an auxiliary function, which maximizes the log-likelihood of the adaptation data given the statistics of the old data, is optimized:

$$\begin{aligned}
\mathcal{Q}_{\mathrm{G}} &= \sum_{t,m}^{T,M} \gamma_m(t)[\log \mathcal{N}(\mathbf{W}\xi_t|\mu_m, \mathbf{\Sigma}_m) + \log(\mathbf{A})] \\
&= \log|\det(\mathbf{A})| \\
&\quad -\frac{1}{2}\sum_{t,m}^{T,M} \gamma_m(t)(\mathbf{W}\xi_t - \mu_m)^T \mathbf{\Sigma}_m^{-1}(\mathbf{W}\xi_t - \mu_m),
\end{aligned}$$

with μ and $\mathbf{\Sigma}$ being the statistics of the training data assumed to follow a distribution which can be modeled as a mixture of M normal distributions, i.e., mean and variance of the Gaussian mixture component m. The posterior probability of sample $\mathbf{x_t}$ belonging to Gaussian mixture component m is depicted by γ. M is the number of mixture components and T the number of samples. By using the sufficient statistics $\mathbf{k}_i$ and $\mathbf{G}_i$:

$$\begin{aligned}
\mathbf{k}_i &= \sum_{t,m}^{T,M} \frac{1}{\sigma_{im}^2}\gamma_{mt}\mu_{im}\xi_t \\
\mathbf{G}_i &= \sum_{t,m}^{T,M} \frac{1}{\sigma_{im}^2}\gamma_{mt}\xi_t\xi_t^T
\end{aligned}$$

the auxiliary function can be reformulated to:

$$\mathcal{Q}_{\mathrm{G}} = \log(|\det(\mathbf{A})|) - \sum_{i=1}^{I}(\mathbf{w}_i\mathbf{k}_i - 0.5\mathbf{w}_i^T\mathbf{G}_i\mathbf{w}_i), \tag{2}$$

with $\mathbf{w}_i$ being the transposed rows of $\mathbf{W}$ and I being the number of dimensions.

3.2. Feature-space Adaptation to the SVM boundary (fSVMA)

Since it is questionable if the data is well described by the superposition of a small number of Gaussian kernels in a Gaussian mixture model and the fMLLR is not well suited for a discriminative classifier, we propose to adapt the data more directly to the trained SVM model. We refer to this as feature-space Support Vector Machine Adaptation (fSVMA). For calculating the SVM model we use the radial basis function kernel (RBF). The RBF is defined through the support vectors $\mathbf{s}$ and the scale parameter γ_{SVM}. Hence the function we want to maximize for calculating the transformation matrix $\mathbf{W}$ is:

$$\mathcal{Q}_{\mathrm{SVM}} = \sum_{t,n}^{T,N} y_t \alpha_n e^{(-\gamma_{\mathrm{SVM}} \|\mathbf{W}\xi_t - \mathbf{s}_n\|^2)}, \tag{3}$$

with the parameters from the learned SVM model, α_n being the weights for each training example, y_i the class affiliation, and N the number of support vectors. The function is optimized using gradient ascent. We iteratively adapt the rows $\mathbf{w}_i$ of $\mathbf{W}$. The first derivative of $\mathcal{Q}_{\mathrm{SVM}}$ with respect to $\mathbf{w}_i$ is:

$$\frac{\partial \mathcal{Q}_{\mathrm{SVM}}}{\partial \mathbf{w}_i} = -2\gamma_{\mathrm{SVM}} \sum_{t,n}^{T,N} \xi_t y_t \alpha_n (\mathbf{w}_i \xi_t - \mathbf{s}_{n,i}) e^{-\gamma_{\mathrm{SVM}} \|\mathbf{W}\xi_t - \mathbf{s}_n\|^2}.$$

The correct choice of the step size is crucial. Therefore, we start with a comparably high step size and decrease it every time $\mathcal{Q}_{\mathrm{SVM}}$ increases. The iteration of the rows $\mathbf{w}_i$ is done until the gain of $\mathcal{Q}_{\mathrm{SVM}}$ between the different iteration steps is very small, defining a border variable ϵ:

$$\mathcal{Q}_{\mathrm{SVM}}(\mathrm{iteration}+1) - \mathcal{Q}_{\mathrm{SVM}}(\mathrm{iteration}) < \epsilon, \qquad \epsilon > 0.$$

If the change is smaller than ϵ, convergence is reached and the iteration is stopped.

3.3. Gaussian-regularized feature-space SVM Adaptation (GrfSVMA)

The estimation of a model with a lot of free parameters and not much available data is difficult and will often lead to overfitting. To control the overfitting, an often used technique is regularization [35]. Usually a penalty term is added to the error function in order to discourage the coefficients from, e.g., reaching large values, to keep the model from getting too complex or to constrain it to bounds or smoothness. A simple example for such a penalty term is to take the sum of squares. To control the influence of the regularization on the adaptation the coefficient λ is used.

Therefore, we propose to regularize the SVM adaptation with the fMLLR, in order to get a tradeoff between the fitting of the adaptation data to the SVM boundary and the more general fitting to the assumed Gaussian distribution. We call this method Gaussian-regularized feature-space SVM Adaptation (GrfSVMA). In the new optimization function the regularization term is weighted by λ_G:

$$\mathcal{Q}_C = \mathcal{Q}_{\mathrm{SVM}} + \lambda_G \mathcal{Q}_{\mathrm{G}}. \tag{4}$$

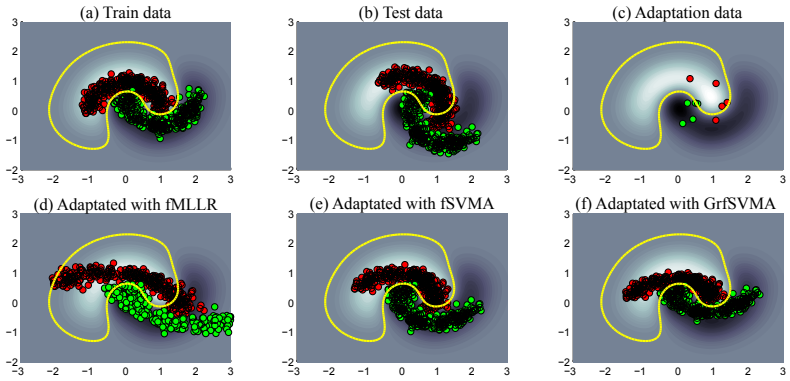


Figure 1: (a) Training data, (b) test data, (c) adaptation data, (d) data after transformation calculated with fMLLR, (e) data after transformation calculated with the fSVMA, (f) data after transformation calculated with the GrfSVMA.

This combined term is also optimized by using gradient descent. Therefore, the partial derivative for $\mathcal{Q}_{\mathrm{G}}$ is:

$$\frac{\partial Q_G}{\partial \mathbf{w}_i} = \beta(\mathbf{W}[\mathbf{c}_i \;\; 0]^T)^{-1}[\mathbf{c}_i \;\; 0]^T + \mathbf{k}_i - \mathbf{G}_i\mathbf{W}^T,$$

where β is the total count and $\mathbf{c}_i$ is the transpose of the rows of the cofactors of A.

3.4. Sparseness Regularization

In the case of speaker adaptation a common approach to reduce the number of free parameters and hence to avoid overfitting is to restrict the transformation matrix to be diagonal. However, this only allows us to scale the data and not to rotate it. An alternative is a least squares regularization. Yet this would minimize all values. A more preferable outcome is a sparse matrix where the larger values, typically the diagonal elements, remain mostly unchanged and the smaller elements, which are typically the off-diagonal elements, are further minimized. Therefore, we choose the L_1-norm. The cost function we optimize is thus:

$$\mathcal{Q}_C = \mathcal{Q}_{\mathrm{SVM}} + \lambda_G\mathcal{Q}_{\mathrm{G}} + \lambda_S\mathcal{Q}_{\mathrm{S}}, \tag{5}$$

with

$$\mathcal{Q}_{\mathrm{S}} = \sum_{i,j}^{I,J} \|w_{i,j}\|, \tag{6}$$

and its derivative:

$$\frac{\partial Q_S}{\partial \mathbf{w}_i} = \sum_{i,j}^{I,J} \mathrm{sign}(w_{i,j}).$$

training data	test data	Adaptation data per class	fMLLR	fSVMA	GrfSVMA
Error [%]			Error [%]		
0.5	16.1	5	12.9	1.8	3.4
		10	5.2	1.2	2.9

Table 1: Error for classification with the training data, test data, adapted data with fMLLR, fSVMA model and GrfSVMA, respectively.

3.5. Artificial data example

To illustrate how the adaptations are working we show the transformation results using the example of the two dimensional classification problem from [36] with two intertwining moon distributions, each belonging to a specific class. Fig. 1 depicts the two classes (green/red) as well as the SVM decision boundary (yellow). 1(a) shows the data used for training, while 1(b) shows the slightly shifted and rotated (45°) data for testing. This example represents a case where an adaptation is useful but the Gaussian assumption does not hold. Especially, the estimation of the variance is difficult if only a small amount of adaptation data 1(c) is available. The sub images 1(d), 1(e) and 1(f) show the test data after transformation with the fMLLR, fSVMA and GrfSVMA ($\lambda_G = 1$). The corresponding classification results can be found in Tab. 1. It is obvious that the adaptation with fMLLR yields only a small gain while there is a risk that the data is stretched too much, to fit to the Gaussian assumption. The SVM based adaptation as well as the method regularized by the fMLLR work quite well and can reach nearly the same performance as on the training data. Here the fSVMA works best, while the additional Gaussian regularization leads to slightly worse results since it also tends to stretch the data.

Tab. 2 shows the results when the data points which were previously used for adaptation are combined with the original training data and the model was then trained on all available data. As can be seen, the results just slightly improve due to the small amount of adaption data, but also due to the compromise the model has to make in areas where one class of the training data and the other class of the test data overlap. This is a rather simple but therefore illustrative example.

train data	test data	test data, retrained model, data points/class:	
		5	10
0.5%	16.1%	15.1 %	15.9%

Table 2: Error for classification with the training data, test data with the original trained model and test data with the retrained model using the 5 and 10 adaptation data points per class in addition to the training data.

To show how the different methods behave if the problem gets more complicated, we did the same experiment with a higher number of dimensions, repeating the moon shaped distribution for every two following dimensions. Here the test data

number of dimensions	training data	test data	fMLLR	fSVMA	GrfSVMA
			Error [%]		
2	2.4	10.0	17.2	1.9	3.3
3	2.2	17.8	16.8	3.7	9
4	1.9	18.1	9.8	30.8	3.6
5	2.1	17.0	8.5	32.1	3.3
10	1.9	26.8	7.5	31.2	1.2

Table 3: Error for classification with the training data, test data, adaptation with fMLLR, fSVMA and GrfSVMA, for 5 adaptation data points per class and different number of dimensions.

.

were transformed by a rotation matrix that rotates dimensions pairwise around 25° and introduces a small bias. From Tab. 3 it can be seen that for higher dimensions the fSVMA gets more unstable, even increases the error. Meanwhile the GrfSVMA yields stable (and better than the fMLLR), always improving results. This indicates that for a more complex and high dimensional problem like the speaker adaptation for word prominence detection, the regularization might prove beneficial.

4. Data set

The audio-visual data used for the following experiments was recorded as a Wizard-of-Oz experiment during a game where the participants moved tiles to uncover a cartoon [37]. We decided to record a dataset on our own and not use an existing one since we mainly wanted to investigate how corrections can be identied based on acoustic features only. Hence we designed our corpus in such a way that the lexical meaning of the words shall not interfere with the prominence realization. Most existing datasets labeled for prominence have the disadvantage for our case that the utterance context has a strong inuence on the prominence realization or that there is not enough data per speaker availabel. Additionally, available corpora are also not based on naturalistic speech but on speech collected from professional news announcers [ref to Boston Radio News Corpus]. The participants had to give spoken advice, in a simple grammar, to a computer, such as: "place green in B one". Some of the words were misunderstood by the machine; this was verbally and visually shown. Then the participants had to correct the sentence by repeating it and using prosodic cues to emphasize the misunderstood word as they would do with a human listener. However, correcting cue phrases such as "no I said" were not allowed. With this procedure we hoped to stimulate a rather natural use of prosody, creating a narrow focus condition (in contrast to the broad focus condition of the original utterance) with the corrected words marked as highly prominent.
For the experiments 16 participants, male and female, speaking either British

or American English as native language or being either bilingual in British English/German or American English/German were recorded. Over 2500 utterances with 4 to 5 words each were used for evaluation. For the experiments in this work we only used the audio recordings. The speech was recorded with a distant microphone at 48 kHz and later downsampled to 16 kHz. A speech recognition system trained on the Grid Corpus [38] was used for forced alignment on the word level.

Three human annotators rated the recorded data with a 4 level scale (0-3) of prominence for each word. In previous experiments a prominence scale with 4 levels had been found to be well adapted [39]. All annotators were trained English teachers and as such had received linguistic training. The presentation of the utterances was randomized along the time and speaker dimension for the annotation. Unfortunately, there is no generally agreed upon prominence scale. In an effort to make the ratings nevertheless consistent, we manually extracted four sentences which could serve as an example of a sentence with a word corresponding to one of the four levels of prominence (i.e. sentence one had a word with prominence level 3, sentence 2 with prominence level 2 ...). We made these sentences together with the labels available to the annotators during their annotation such that they could use them as a reference. We calculated the inter-annotator agreement with Fleiss' kappa κ. While doing so we binarized the annotations, i.e., only differentiating between prominent and non-prominent. We tested different binarizations and used the one where the agreement between all annotators was highest. Next, we calculated κ for each speaker individually. We then discarded all speakers where κ for the optimal binary annotation was below 0.55 ($0.4 < \kappa \leq 0.6$ is usually considered as moderate agreement). This yields a data set of 8 speakers. Of those remaining eight speakers, two are female and six male, two speak American English (including one being bilingual in American English/German) and six speak British English (including two being bilingual in British English/German). In Tab. 4 the number of utterances, prominent words, total words and κ are shown for the individual speakers. Overall we have 4927 utterances with 22496 words of which 1263 are corrections, i.e., on average approx. 616 utterances per speaker with approx. 25.6% corrections. As mentioned previously, our main objective was to acquire data where humans use word prominence to indicate in an interaction a misunderstanding of the system. Unfortunately, the interaction with the wizard system via speech turned out to be rather uncommon and challenging for some of our participants. We attribute the low inter-annotator agreements we see in Tab. 4 mainly to this fact. As a consequence, we had to choose a rather low κ threshold to retain as many speakers as possible.

5. Features

For the detection of word prominence we employ the prosodic features which have been previously proposed to correlate with word prominence (compare Table 5 and Fig. 2 for an overview on all features and functionals we used). The beginning and end of the word is taken from the forced alignment and used to

speaker	number of utterance	number of prominent words	number of all words	κ
1	385	97	1540	0.63
2	697	169	3485	0.62
3	725	181	3625	0.62
4	540	140	2160	0.74
5	673	177	3365	0.60
6	498	146	1992	0.71
7	716	172	2864	0.74
8	693	181	3465	0.56
overall	4927	1263	22496	$\varnothing\kappa = 0.65$

Table 4: Number of utterances, prominent words, total words and κ per speaker as well as overall counts and average, respectively.

used features	functionals	DCT	context
loudness in dB	x	x	x
duration of the word			x
duration of the gap between words			x
fundamental frequency in semitones	x	x	x
spectral emphasis, i.e., the difference between overall energy and low-pass energy			x

Table 5: The used features.

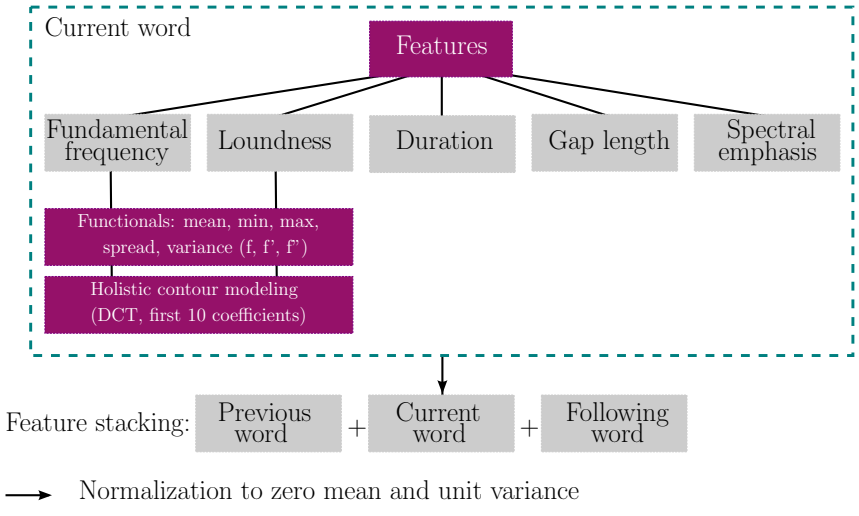


Figure 2: The used features.

calculate the duration of a word as well as the gap length before and after the word.

The loudness is calculated by first filtering the signal with a 12th order IIR filter following the ideas outlined in [40], followed by the calculation of the instantaneous energy, smoothing with a low pass filter with a cut-off frequency of 10 Hz, and finally converting the results to dB. We expected the loudness to better capture the perceptual correlates of prominence than the energy.

As described in [41], we extract the fundamental frequency f_0 (following [42]), interpolate values in the unvoiced regions via cubic splines and convert the results to semitones. To detect voicing we use an extension of the algorithm described in [43]. Another feature we use is the spectral emphasis, i.e., the difference between the overall intensity and the intensity in a dynamically low-pass-filtered signal with a cut-off frequency of 1.5 f_0 [44].

From the fundamental frequency, energy and loudness, we extract functionals as described in [41] for a better prosodic analysis. We extract the mean, max, min, spread (max-min) and variance along the word. Prior to the calculation of the functionals we normalize the prosodic features by their utterance mean and calculate their first and second derivative. To model the contour we calculate the Discrete Cosine Transformation from the features and retain the first 10 coefficients.

Marking the focus of a word in an utterance, rendering it prominent, also has an influence on the neighboring words. Modifying the articulatory parameters to raise the prominence of a segment of an utterance (hyperarticulating) is usually accompanied by a reduction of these parameters (hypoarticulation) for the neighboring segments [45, 46]. It has previously been shown that taking this context information into account is very effective for the detection of word prominence [47] and pitch accents [48, 49]. Pitch accent is frequently used as intonational prominence or, in some languages, lexically to differentiate words. Therefore, we take not only the functionals of the current word itself for classification but also those of the previous and the following word (see [47] for details).

Considering all audio features with their functionals and context features we obtain a feature vector with a dimension of 159.

6. Experimental Setup

As mentioned before, for problems such as word prominence detection, discriminative classifiers proved to be a good choice. Therefore, we use a classification with a support vector machine (SVM). We used the libsvm implementation [50] with a radial basis function kernel. For the parameter C, the penalty parameter of the error term, and the variance scaling factor γ of the basis function, a grid search on every training data set was performed.

Furthermore, for evaluation, we compared the SVM results to those obtained with Gaussian Mixture Models (GMM), since they harmonize well with the MLLR/fMLLR adaptation as these adaptation schemes also use a GMM representation. Additionally, we used Deep Neural Networks (DNN) for classification,

since they proved useful for many different machine learning problems. For the DNN implementation we used Tensorflow [51] using a mini-batch gradient descent with an Adam optimization and a batch size of 100. Following the work in [52], we performed experiments with 1 to 5 hidden layers with 25 to 400 hidden units per hidden layer. To avoid a preference of the dominant class we use an oversampling strategy to balance both classes.

In order to use the adaptation we determined mean and variance of the training data, scaled them to zero mean and unit variance and then used the same scaling factors for the test and the adaptation data.

We use the supervised version for adaptation, taking a small labeled subset of the test data for adaptation. In doing so, we use the same amount of prominent and non-prominent samples to calculate the transformation matrix. We experimented with full covariance matrices [53], but it is a well-known problem that the estimation of the covariance matrix with a small amount of adaptation data is difficult; especially, with fewer samples per class than dimensions [54]. So on the one hand, we assumed a diagonal covariance matrix for the training and the adaptation data. On the other hand, we did a feature reduction with a PCA transformation taking only the first 79 dimensions, because discarding the 80 higher dimensions did not change the performance of the SVM, only for the experiments with DNNs we did not perform the PCA dimension reduction, since with the reduced dimension the performance was slightly worse.

7. Results

Because the two classes, prominent and non-prominent, are highly unbalanced in our data set, approx. one times to nine, the accuracy is not a good measure to compare the results. A preference for non-prominent words would always lead to very good accuracy but would not provide an objective evaluation. Consequently, we use the unweighted error rate instead:

$$\begin{aligned} \text{precision} &= \frac{tp}{tp+fn}, \\ \text{specificity} &= \frac{tn}{tn+fp}, \\ \text{unweighted error} &= 1 - \frac{\text{precision} + \text{specificity}}{2}, \end{aligned}$$

with tp being true positive, fp false positive, tn true negative and fn false negative.

To better understand the possible gain of the speaker adaptation, we will first show the performance difference between the speaker dependent and speaker independent classification. For speaker independent classification the model is trained on 7 speakers and tested on the last one (leave-one-speaker-out classification). For speaker dependent classification the data of each speaker is divided into a training set, containing 75% of the data, and a test set, containing the

	speaker dependent	speaker independent
GMM	17.1%	22.0%
SVM	11.5%	16.4%
DNN	12.1%	16.7%

Table 6: Speaker dependent vs. speaker independent classification with an SVM and a GMM for prominence detection: unweighted error rate average over all speakers.

remaining 25% of the data, using a 30 fold cross validation, in which the data set is always split in a way that the same number of elements is taken from both classes. To establish the 30 sets, a sampling with replacement strategy was applied. The training and testing was done for each speaker independently. Tab. 6 shows the results for speaker dependent classification compared to the speaker independent classification with an SVM, a GMM and a DNN. For all classifiers, the average unweighted error rate over all speakers for speaker dependent training is lower than the speaker independent trained models. The GMM has the highest unweighted error rate, where the speaker dependent classification with 17.1% is already worse than the speaker independent classification with the other two classifiers. If we have a speaker independent classification the result for the averaged unweighted error rate increases to 22%.

The difference between an SVM and a DNN is small with a DNN having a 0.3% higher error rate for speaker independent classification and a 0.6% higher error rate for speaker dependent classification. We see the best performance using an SVM with 16.4% (speaker independent) and 11.5% (speaker dependent) error rate. The performance difference between speaker independently and speaker dependently trained models of up to 4.9% shows that there is potential for a speaker personalization like the proposed feature adaptation.

When investigating the performance of the DNN we varied the number of hidden layers and units per layer between 1-5 and 25-400, respectively. The corresponding error rates for speaker dependent training can be seen in Fig. 3. As a reference the results for the SVM classification are also depicted. The smallest error rate is reached for 4 Layers and 25 Neurons with 12.1%. Fig. 4 depicts the results of the corresponding experiment for the speaker independent classification task. Here the best value is reached for 3 Layers and 25 Neurons with 16.7%.

We also experimented with different numbers of Gaussian components in the mixture, which can be seen in Tab. 7 for one, two and three Gaussian mixture components per class. We determine the probability of belonging to each Gaussian mixture component for each data point, normalizing the overall probability to one and using them as weights. While for one Gaussian mixture component the performance improved through adaptation, for two and three Gaussian mixture components the performance did not change much. For two Gaussian mixture components the performance is 21.6% and for three it is 24.4%.

Since taking more than one Gaussian mixture component per class did not im-

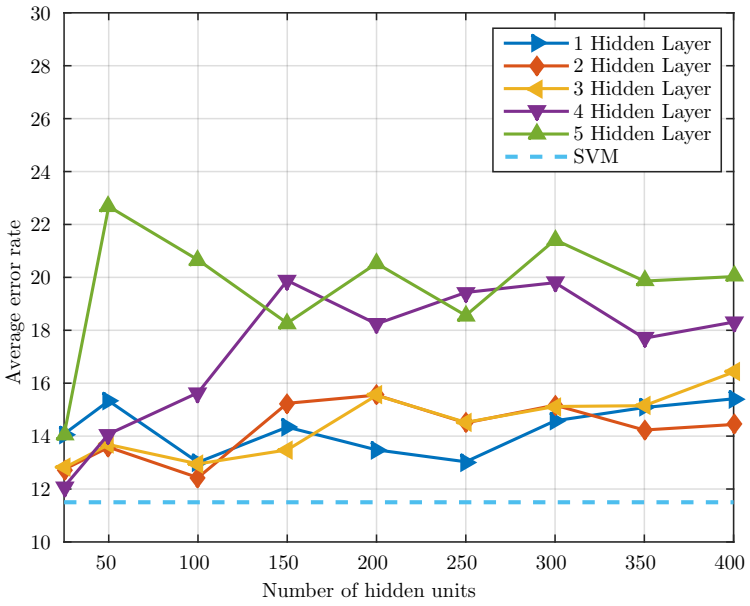


Figure 3: The average unweighted error over all speakers for speaker dependent classification with a DNN for prominence detection, using varying numbers of hidden layers and units.

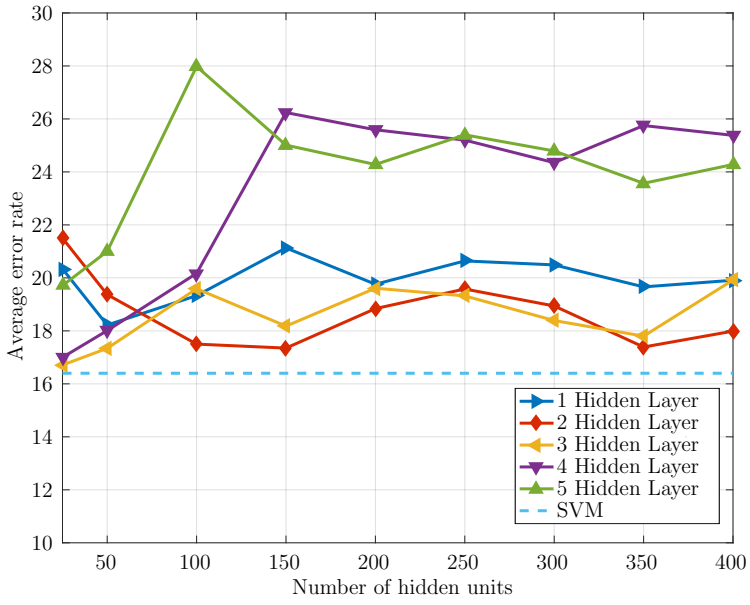


Figure 4: The average unweighted error over all speakers for speaker independent classification with a DNN for prominence detection, using varying numbers of hidden layers and units.

	number of components	without adaptation	with adaptation
GMM	1	22.0%	20.4%
GMM	2	21.6%	21.8%
GMM	3	24.4%	24.2%
SVM	1	16.4%	16.0%
SVM	2	16.4%	16.1%
SVM	3	16.4%	16.1%

Table 7: Unweighted error rate for speaker independent classification with an SVM and a GMM with and without adaptation and different number of Gaussian mixture components [1].

adaptation technique		classifier	adaptation data per class error [%]					
			10	20	30	40	50	100
retrained		SVM	16.4	16.2	16.2	16.1	16	16.1
fMLLR		GMM	22.2	21.0	20.6	20.4	20.3	19.8
fMLLR		SVM	18.1	16.9	16.2	16.0	15.8	15.6
fSVMA	$(\lambda_G : 0\ ,\lambda_S : 0\ )$	SVM	36.4	31.5	28.7	27.5	24.8	19.8
sparse fSVMA	$(\lambda_G : 0\ ,\lambda_S : 5\ )$	SVM	16.4	17.9	18.7	20.6	21.8	18.6
GrfSVMA	$(\lambda_G : 1\ ,\lambda_S : 0\ )$	SVM	17.1	16.0	15.0	14.8	14.3	13.2
sparse GrfSVMA	$(\lambda_G : 1\ ,\lambda_S : 5\ )$	SVM	15.4	15.2	14.7	14.3	14.0	13.4

Table 8: Unweighted error rate for speaker independent classification with an SVM and the different adaptation techniques using varying amounts of adaptation data.

prove the results, we use only one in the following experiments.

In the first experiment, the effect of varying amounts of adaptation data on the different adaptation techniques was investigated. The data of a new speaker were split, such that some data were taken for adaptation and the remaining data were taken for testing.

Tab. 8 shows the results for the speaker independent classification for the different adaptation techniques using either an SVM or a GMM for classification. The first line, termed retrained, depicts the case when we trained the model from scratch using the original data together with the adaptation data. When adding the adaptation data to the training data to retrain the model, there is only a very small improvement since the amount of adaptation data is very small compared to around 10000 original data points for training.

For testing the adaptation we did a 10 fold cross validation, where we split the data of the new speaker in adaptation and testing data, taking the adaptation data to calculate the transformation matrix and the test data for evaluating the performance of the adaptation. For the combination of a GMM and fMLLR

[1] In case of the SVM the number of mixture components only refers to those used during adaptation.

adaptation technique		classifier	speaker error [%]							
			1	2	3	4	5	6	7	8
without adaptation		SVM	17.5	23.2	11.9	17.1	21.9	10.8	13.4	16.1
fMLLR		SVM	18.0	16.6	12.5	19.1	17.5	14.0	12.1	18.1
fMLLR		GMM	26.1	18.8	16.3	19.7	21.1	21.5	18.5	21.4
fSVMA	($\lambda_G : 0$,$\lambda_S : 0$)	SVM	25.6	31.3	20.3	21.0	45.0	21.3	19.2	36.0
sparse fSVMA	($\lambda_G : 0$,$\lambda_S : 5$)	SVM	21.0	20.4	19.3	31.1	24.8	13.2	14.4	20.9
GrfSVMA	($\lambda_G : 1$,$\lambda_S : 0$)	SVM	18.9	14.2	11.3	14.0	16.5	14.6	13.0	16.0
sparse GrfSVMA	($\lambda_G : 1$,$\lambda_S : 5$)	SVM	18.4	13.9	11.0	14.8	16.6	10.9	12.3	16.5

Table 9: Unweighted error rate for every speaker for speaker independent classification with an SVM and the different adaptation techniques for 40 adaptation data points.

adaptation already for 20 data points per class a small improvement can be seen. For 40 data points the error decreases from 22.0% to 20.4% and to 19.8% for a large amount of adaptation data.
Using fMLLR for 40 data points for an SVM classification the performance did not change much, but for a large amount of adaptation data the error decreases slightly to 15.6%. When using only the adaptation to the SVM model, the fSVMA, the performance decreases in all cases, but the error decreases with increasing amounts of adaptation data. For the GrfSVMA, with 40 data points we already reduce the error to 14.8%, with 50 data points we further decrease it to 14.3% and with 100 data points the error is only 13.2%.
For fSVMA using the L_1-norm (λ_G:0 ,λ_S:5), here called sparse fSVMA, and 10 data points there is no decrease or increase of the error. With increasing amounts of adaptation data the error further increases, but being always smaller than without the sparseness. For the sparse GrfSVMA (λ_G:1 ,λ_S:5) already with only 10 data points per class the error decreases to 15.4% and further decreases with higher amounts of adaptation data. The error is again smaller than without the sparseness term, converging to the same result, when having high amounts of adaptation data.
For better comparison the results of the different adaptation methods used with an SVM classification are plotted in Fig. 5 and, when using the additional sparseness term, in Fig. 6. Note that the small fluctuations for the baseline are due to the changing amount of testing data (all data of the new speaker minus adaptation data) which is used.

The results for the different adaptation methods for every speaker individually are shown in Tab. 9. Without the adaptation the results for SVM-based classification vary between 10.8% to 23.2.%. For three speakers the results after adaptation are slightly worse than prior to the adapation. The highest improvement is reached for the two speakers with the highest error prior to the adapation (speakers two and five). In particular, the highest improvement ist reached for speaker two with 9.3% improvement. When looking at Tab. 9 one can also observe that for speakers four, five and eight the sparseness term seems

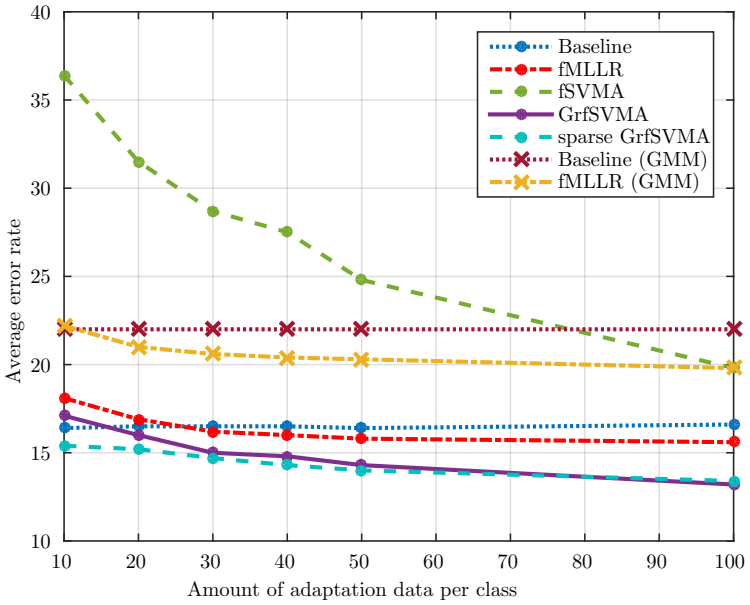


Figure 5: Unweighted error rate for speaker independent classification with an SVM and a GMM and the different adaptation techniques using varying amounts of adaptation data.

not to be optimally adapted as the results with the additional sparseness term are worse than without it.

To further evaluate the performance of our adaptation method we used different values for λ_G to control the Gaussian regularization and for λ_S to control the sparseness. The results of this experiment are shown in Tab. 10. We set the value λ_G between 0 (meaning no Gaussian regularization) and 5 and for λ_S between 0 (no sparseness regularization) and 10. Fig. 7 illustrates the influence of the parameter choice on the outcome. The smallest errors are in dark blue. The largest change takes place for $\lambda_G = 0$ to $\lambda_G = 1$, showing how important this regularization term is. The worst result occurs when no regularization term at all is used. Considering only the Gaussian regularization the best results are obtained for $\lambda_G = 1$ respectively $\lambda_G = 3$ for a smaller amount of adaptation data. Considering only the sparseness term the error decreases compared to the results without the term, but is still higher than the baseline. But with a large value for λ_S the transformation matrix does not change during optimization and remains an identity matrix. Consequently the features are not transformed and the error can not increase. This is the more so the less adaptation data is available which can be seen especially in Fig. 7(a). Combining both regularization terms leads to the best results. Regarding the three cases (10, 40 and 100 data points per class) Fig. 7 shows that the best results (the dark blue tiles) shifts from right (a) to left (c) showing that the impact of the sparseness regularization decreases the more adaptation data is available. Thus considering

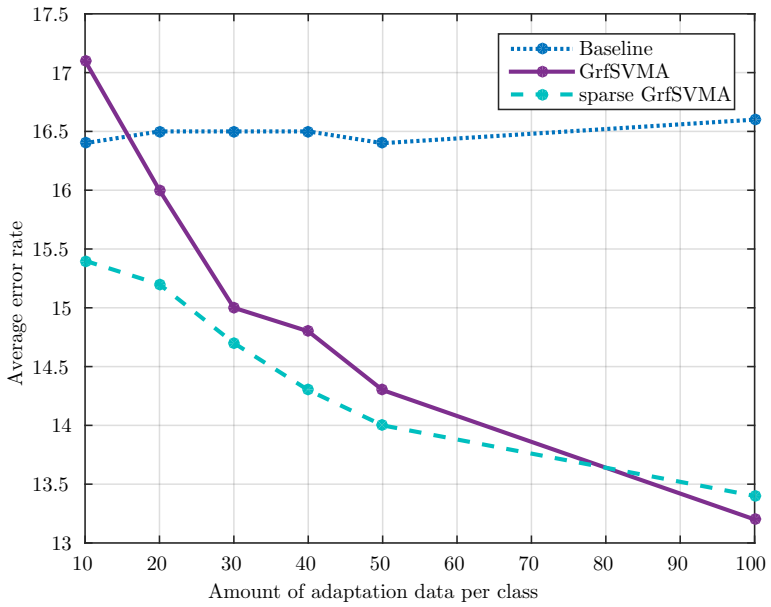


Figure 6: Unweighted error rate for speaker independent classification with an SVM and the different adaptation techniques with and without sparseness term, using varying amounts of adaptation data.

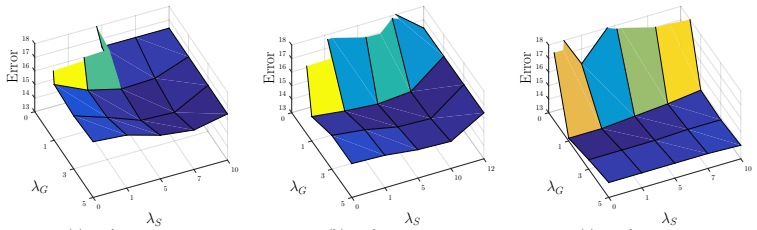
$\lambda_G = 1$ (for 40 and 100 data points) or $\lambda_S = 3$ (10 data points), the best values range from $\lambda_S = 7$ (10), $\lambda_S = 5$ (40) to $\lambda_S = 0$ or $\lambda_S = 1$ (100). Using these parameters leads to an improvement of 1.3% to 3.2%, which Tab. 10 presents in detail.

8. Discussion

Due to the high inter-speaker variations the unweighted error rate averaged over all speakers increases from 11.5% for speaker dependent to 16.4% for speaker independent training with an SVM, from 12.1% to 16.7% for a DNN and from 20.4% to 22.0% for a GMM, although the amount of training data is much higher for the independent model. That underlines that the individual speakers differ from each other and a speaker independent model should benefit from an adaptation to each speaker. Our experiments illustrate that for this problem and feature choice, a DNN performs quite similarly to an SVM, while a GMM has a higher error rate. Adding more hidden layers to the DNN was shown to be beneficial for the speaker dependent case as well as for the speaker independent case, where the optimum is reached for 3 and 4 hidden layers, respectively. As for the number of neurons, a smaller number was better for our problem case. This might be due to the rather small database we use for training. Using more than 4 hidden layers and a higher number of neurons led to an increase of the

	amount of adaptation data per class				
			10		
	λ_S: 0	λ_S: 1	λ_S: 5	λ_S: 7	λ_S: 10
λ_G: 0	36.4%	26.6%	16.4%	16.4%	16.4 %
λ_G: 1	17.1%	16.0%	15.4%	15.5%	16.4 %
λ_G: 3	16.9%	16.1%	15.2%	15.1%	16.4 %
λ_G: 5	17.1%	16.9%	16.2%	15.9%	16.4 %
			40		
	λ_S: 0	λ_S: 1	λ_S: 5	λ_S: 7	λ_S: 10
λ_G: 0	27.5%	18.6%	20.6%	18.3%	16.4%
λ_G: 1	14.8%	14.5%	14.3%	14.5%	16.4%
λ_G: 3	15.5%	15.2%	14.7%	14.6%	16.4%
λ_G: 5	15.5%	15.4%	15.1%	15.0%	16.4%
			100		
	λ_S: 0	λ_S: 1	λ_S: 5	λ_S: 7	λ_S: 10
λ_G: 0	19.8%	15.9%	18.6%	20.9%	21.6%
λ_G: 1	13.2%	13.2%	13.4%	13.6%	13.9%
λ_G: 3	13.7%	13.7%	13.7%	13.8%	13.8%
λ_G: 5	14.1%	14.1%	14.1%	14.1%	14.0%

Table 10: Error for classification with the sparse GrfSVMA, using different λ_G and λ_S.



(a) 10 data points (b) 40 data points (c) 100 data points

Figure 7: Error for classification with the sparse GrfSVMA, using different λ_G and λ_S.

error.

Overall the SVM performed a little better than the DNN. The classification with our real-world data is rather complex due to the high dimensionality and the overlapping classes. Using fMLLR with 40 data points per class to calculate the transformation matrix, did not lead to a notable improvement for an SVM while for a GMM an improvement could be reached. This shows that the adaptation works in general on the data, but is not useful in combination with a discriminative classifier like an SVM.

Our newly proposed adaptation method achieves a notable improvement also for the SVM. The results highlight that the parameter setting as well as the amount of adaptation data plays an important role. Regarding the two regularization terms, the experiments illustrated that the worst results occur without any regularization and the best with a combination of both. Using only the adaptation to the SVM boundary (both regularization terms set to 0) did increase the error. The reason might be that with the small amount of adaptation data, a strong overfitting to the adaptation data occurs. This is one of the main problems adaptation techniques have to solve. For such a high dimensional feature space, a large number of parameters has to be estimated on as little as possible adaptation data. The two regularization terms we introduced help to achieve this goal.

Using just the Gaussian regularization improves the results compared to the baseline. But via the Gaussian regularization without the sparseness term, an improvement can only be reached for a higher amount of adaptation data. For a slight improvement at least 20 data points are needed. Applying the sparseness term without the Gaussian term improves results compared to λ_G=λ_S=0 but not compared to the baseline. When properly adjusting both terms, already for 10 data points a notable improvement is reached.

In general we observed that a suitable setting of the regularization terms depends on the amount of available adaptation data. For 10 data points $\lambda_G = 3$ is optimal. When 40 or 100 data points are available $\lambda_G = 1$ yields the best results. Similarly, for 10 data points $\lambda_S = 7$ is best. With 40 or 100 data points $\lambda_S = 5$ and $\lambda_S = 1$ are optimal, respectively. Hence the more data is available the smaller the regularization terms should be chosen Both parameters have only a small influence on each other. It is not possible to compensate one through a stronger weighting of the other. The results in Tab. 9 showed that our approach does not show the same level of effectivness for all speakers we investigated. Based on the large differences in performance between the speakers already prior to the adapation this was to be expected. However, the results also indicate that the value of the regularization parameters are not optimal for all speakers, e.g., the sparsness term deteriorated the results for some speakers. From this we conclude that it will be beneficial to go in the adaptation to the individual speaker one step further and also adapt the regualrization terms to the individual speaker.

In our experiments we can see, that with this novel adaptation method, we obtain improvements compared to the baseline even when using 10 data points for the adaptation. Overall the sparse GrfSVMA leads to an improvement from

method	amount of adaptation data	best absolute error[%]	relative improvement [%]
GrfSVMA	10	16.9	-3.0
	40	14.8	9.8
	100	13.2	19.5
sparse GrfSVMA	10	15.1	7.9
	40	14.3	12.8
	100	13.2	19.5

Table 11: Error rates and relative improvements for GrfSVMA and sparse GrfSVMA for best parameter combinations.

16.4% to 15.1% with 10 data points per class, 14.3% with 40 and 13.2% with 100. In Tab. 11 the best results concerning the different parameter choices are shown for the GrfSVMA ($\lambda_S = 0$,$\lambda_G > 0$) and the sparse GrfSVMA ($\lambda_S > 0$,$\lambda_G > 0$) as well as the relative improvements compared to the results without adaptation. Both methods reach a relative improvement of up to 19.5%. But especially the sparse GrfSVMA can already with a low amount of adaptation data reach a relative improvement of 7.9%.

9. Conclusion

In this paper we presented a new method for feature-space SVM adaptation with RBF kernel and evaluated it on the problem of speaker adaptation for word prominence detection. To our knowledge there is no previous work on speaker adaptation methods for the detection of prosodic features; especially, not for a classification with an SVM. Our experiments showed that the adaptation of a discriminative classifier like the SVM is sensible, since for problems like our case, the results of the SVM prior to adaptation are already better than those of a GMM after adaptation. Additionally we saw in our experiments that using a DNN with several hidden layers and some parameter tuning, a similar but not superior performance than with an SVM can be reached. Hence, for problems where not much training data is available, like in our case, methods like SVMs can be a better choice. The large difference between speaker dependent and speaker independent detection illustrated the potential of speaker adaptation. We perfomred our experiments on a dataset where speakers realized word prominence independent on the lexical content. This allowed us to investigate the effects of the acoustic features alone. Even though we made during the recording of our dataset an attempt to elicit a natural use of word prominence, our wizard of Oz setting shows clear differences to real conversational speech. Hence, we can not say with certainty that the conclusions we draw will generalize to other situations.

In the case of the word prominence detection, where it can be expected that only little adaptation data is available in comparison to the feature dimensionality, the adaptation to the decision boundary of the SVM leads to an overfit-

ting. Nevertheless, we were able to show that adding a Gaussian regularization, GrfSVMA, leads to a reduction of the error rate from 16.4% to 16.0%-13.2% (depending on the amount of adaptation data). However, we only observed an improvement when at least 20 data points for adaptation were available. With the additional sparseness term, already for 10 data points the error rate improved to 15.1% and yielded further improvements when more adaptation data was available. Overall we see improvements through adaptation to the speaker in the range of 7.9% to 19.5% relative.

Acknowledgment

We thank Volker Willert, Heiko Wersing, Lydia Fischer, Nikola Aulig and Viktor Loosing for fruitful discussions. Many thanks to Merikan Koyun for his help in the data preparation.

References

[1] E. Shriberg, Spontaneous speech: How people really talk and why engineers should care, in: Proc. European Conf. Speech Communication and Technology (Eurospeech), Lisboa, Portugal, 2005, pp. 1781–1784.

[2] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van, E. Meteer, Dialogue act modeling for automatic tagging and recognition of conversational speech, Comput. Linguist. 26 (2000) 339–373.

[3] J. Hirschberg, D. Litman, M. Swerts, Characterizing and predicting corrections in spoken dialogue systems, Comput. Linguist 32 (2006) 417–438.

[4] A. Rosenberg, Automatic detection and classification of prosodic events, ph.d. dissertation, Ph.D. thesis, Columbia University, NY (2009).

[5] B. Streefkerk, Acoustical correlates of prominence: A design for research, in: Proc. Institute of Phonetic Sciences of the University of Amsterdam, Amsterdam, 1997.

[6] J. Hirschberg, Litman, D., M. Swerts, Prosodic and other cues to speech recognition failures, Speech Communication 43 (2004) 155–175.

[7] M. Swerts, D. Litman, J. Hirschberg, Corrections in spoken dialogue systems., in: Proc. Interspeech, Beijing, China, 2000, pp. 615–618.

[8] M. Gregory, Y. Altun, Using conditional random fields to predict pitch accents in conversational speech, in: Proc. 42nd Meeting Association for Computational Linguistics (ACL'04), Barcelona, Spain, 2004, pp. 677–683.

[9] G. Levow, Automatic prosodic labeling with conditional random fields and rich acoustic features, IJCNLP (2008) 217–224.

[10] H. Moniz, A. Mata, J. Hirschberg, F. Batista, A. Rosenberg, I. Trancoso, Extending autobi to prominence detection in european portuguese, in: Proc. Speech Prosody, 2014.

[11] A. Rosenberg, R. Fernandez, B. Ramabhadran, Modeling phrasing and prominence using deep recurrent learning, in: Sixteenth Annual Conference of the International Speech Communication Association, 2015.

[12] J. Jeon, Y. Liu, Automatic prosodic event detection using a novel labeling and selection method in co-training, Speech Communication 54 (3) (2012) 445–448.

[13] F. Tamburini, C. Caini, An automatic system for detecting prosodic prominence in american english continuous speech, International Journal of speech technology 8 (2005) 33–44.

[14] O. Kalinli, S. Narayanan, A saliency-based auditory attention model with applications to unsupervised prominent syllable detection in speech, in: Eighth Annual Conference of the International Speech Communication Association, 2007.

[15] S. Kakouros, O. Räsänen, 3pro–an unsupervised method for the automatic detection of sentence prominence in speech, Speech Communication 82 (2016) 67–84.

[16] A. Schnall, M. Heckmann, Comparing speaker independent and speaker adapted classification for word prominence detection, in: Proc. IEEE Spoken Language Technology Workshop (SLT), San Diego, California, USA, 2016.

[17] M. Gales, Maximum likelihood linear transformations for hmm-based speech recognition, Computer Speech and Language 12 (1998) 75–98.

[18] A. Schnall, M. Heckmann, Speaker adaptation for support vector machine-based word prominence detection, in: Proc. Speech Prosody, Boston, MA, USA, 2016.

[19] A. Schnall, M. Heckmann, Balancing gaussianity and sparseness in feature-space speaker adaptation for word prominence detection, in: Proc. ITG Conf. Speech Communication, Paderborn, Germany, 2016.

[20] E. Eide, H. Gish, A parametric approach to vocal tract length normalization, in: Proc. ICASSP, Atlanta, Georgia, USA, 1996.

[21] J. Gauvain, C. Lee, Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains, IEEE Trans. Speech and Audio Process. 2 (1994) 291–298.

[22] B. Schuller, Speaker, noise, and acoustic space adaptation for emotion recognition in the automotive enviroment, in: Proc. ITG-Fachtagung Sprachkommunication, Aachen, Germany, 2008.

[23] S. Ananthakrishnan, S. Narayanan, Unsupervised adaptation of categorical prosody models for prosody labeling and speech recognition., IEEE Trans. Audio, Speech and Language Process. 17 (2009) 138–149.

[24] J. Yang, R. Yan, A. Hauptmann, Cross-domain video concept detection using adaptive svms, in: Proc. 15th ACM Int. Conf. Multimedia,Augsburg, Germany, ACM, 2007.

[25] N. Cristianini, C. Campbell, J. Shawe-Taylor, Dynamically adapting kernels in support vector machines, in: Proc. Conf. Advances in Neural Information Processing Systems II, Cambridge, MA, USA, 1999.

[26] G. Cauwnberghs, T. Poggio, Incremental and decremental support vector machine learning, in: Proc. Advances in Neural Information Processing Systems (NIPS), Denver, USA, 2001.

[27] C. Diehl, G. Cauwenberghs, Svm incremental learning, adaptation and optimization, in: Proc. IJCNN, Portland, Oregon, USA, 2003.

[28] S. Yu, X. Yang, Z. Hao, Y. Liang, An adaptive support vector machine learning algorithm for large classification problem, in: Advances in Neural Networks, Springer Berlin Heidelberg, 2006.

[29] V. Losing, B. Hammer, H. Wersing, Choosing the best algorithm for an incremental on-line learning task, in: Proc. European Symp. on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium, 2016.

[30] H. Daumé, III, D. Marcu, Domain adaptation for statistical classifiers, Journal Artificial Intelligence Research 26 (2006) 101–126.

[31] L. Bruzzone, M. Marconcini, Domain adaptation problems: A dasvm classification technique and a circular validation strategy, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 770–787.

[32] W. Jiang, E. Zavesky, S. Chang, A. Loui, Cross-domain learning methods for high-level visual concept classification, in: Proc. 15th IEEE Int. Conf.Image Processing, San Diego, CA, USA, 2008.

[33] L. Duan, I. Tsang, D. Xu, S. Maybank, Domain transfer svm for video concept detection, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition, Miami Beach, FL, USA, 2009.

[34] K. Sim, M. Gales, Adaptation of precision matrix models on large vocabulary continuous speech recognition, in: Proc. ICASSP, Philadelphia, PA, USA, 2005.

[35] C. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[36] Y. Ma, G. Guo, Support Vector Machines Applications, Springer Int. Publishing, 2014.

[37] M. Heckmann, Audio-visual evaluation and detection of word prominence in a human-machine-interaction scenario, in: Proc. Interspeech, Portland, OR, 2012.

[38] M. Cooke, J. Barker, S. Cunningham, X. Shao, An audio-visual corpus for speech perception and automatic speech recognition, The Journal Acoustical Society of America 120 (2006) 2421–2424.

[39] C. Jensen, J. Tøndering, Choosing a scale for measuring perceived prominence, in: Ninth European Conference on Speech Communication and Technology, 2005.

[40] Replaygain, 1.0 specification, `http://wiki.hydrogenaudio.org/`.

[41] M. Heckmann, Audio-visual word prominence detection from clean and noisy speech, Computer Speech and Language 48 (2018) 15–30.

[42] M. Heckmann, F. Joublin, C. Goerick, Combining rate and place information for robust pitch extraction, in: Proc. Interspeech, Antwerp, 2007.

[43] T. Kristjansson, S. Deligne, P. Olsen, Voicing features for robust speech detection, in: Proc. Interspeech and Eurospeech, Lisboa, Portugal, 2005.

[44] M. Heldner, On the reliability of overall intensity and spectral emphasis as acoustic correlates of focal accents in swedish, Journal of Phonetics 31 (2003) 39 –62.

[45] Y. Xu, C. Xu, Phonetic realization of focus in english declarative intonation, Journal of Phonetics 33 (2005) 159–197.

[46] M. Dohen, H. Loevenbruck, Interaction of audition and vision for the perception of prosodic contrastive focus, Language and speech 52 (2009) 177–206.

[47] A. Schnall, M. Heckmann, Integrating sequence information in the audio-visual detection of word prominence in a human-machine interaction scenario, in: Proc. Interspeech, Singapour, 2014.

[48] G. Levow, Context in multi-lingual tone and pitch accent recognition, in: Proc. Interspeech, Lisboa, Portugal, 2005.

[49] A. Rosenberg, J. Hirschberg, Detecting pitch accents at the word, syllable and vowel level, HLT-NAACL, Boulder, Colorado (2009) 81–84.

[50] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Trans. Intelligent Systems and Technology 2 (2011) 1–27, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[51] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, `http://tensorflow.org/`, software available from tensorflow.org (2015).

[52] M. Shahin, B. Ahmed, K. Ballard, Classification of lexical stress patterns using deep neural network architecture, in: Proc. IEEE Spoken Language Technology Workshop (SLT), South Lake Tahoe, Nevada, USA, 2014.

[53] D. Povey, G. Saon, Feature and model space speaker adaptation with full covariance gaussians., in: Proc. Interspeech, Pittsburgh, PA, USA, 2006.

[54] J. Hoffbeck, D. Landgrebe, Covariance matrix estimation and classification with limited training data, IEEE Trans. Pattern Anal. Mach. Intell. 18 (1996) 763–767.