COMPARISON OF CURRENT FRAME-BASED PHONEME CLASSIFIERS

Vaclav PFEIFER[1], *Miroslav BALIK*[1]

[1]Department of Telecommunications, Faculty of Electrical Engineering and Communication, Brno University of Technology, Purkynova 118, 612 00 Brno, Czech Republic

pfeifer@feec.vutbr.cz, balik@feec.vutbr.cz

Abstract. *This paper discusses current approaches for frame-based classification and evaluates today's most common frame-based classifiers. These classifiers can be divided into the two main groups – generic classifiers which create the most probable model based on the training data (for example GMM) and discriminative classifiers which focus on creating decision hyper plane (SVM based methods). A lot of research has been done with the generic classifiers and therefore this paper will be mainly focused on the discriminative classifiers. Four discriminative classifiers are presented – two linear and two non-linear. All of these discriminative classifiers implement a hierarchical tree root structure over the input phoneme group which shown to be an effective. Moreover, two efficient training algorithms are presented. First, we demonstrate advantages of discriminative classifiers by comparison with a standard generic classifier represented by a GMM. Second, we show benefits of our proposed training algorithm. All tests were performed for English only - over the TIMIT speech corpus (corpus of Native American speakers).*

Keywords

Classifier, comparison, frame-based, phoneme, speech, hierarchical.

1. Introduction

Phoneme classification is a task of deciding the identity of an unknown speech utterance (mostly short ones) [1]. The correct classification plays important role in most of the current state-of-the-art speech systems – for example speech recognition, spoken term detection, etc. Based on the classifier input, the classifiers can be further divided into sequence based classifiers and frame based classifiers. Most of the current speech processing systems are based on the sequence based classifiers. The sequence modeling is performed using a Hidden Markov modeling (HMM) [2], [3] and classification itself can be done using Gaussian mixture modeling (GMM) [4], neural network (NN) [5] or support vector machines (SVM) [6]. These systems are mostly denoted as a HMM/GMM or HMM/NN, etc. [4], [7], [8]. The main advantages of these systems are simple phoneme modeling and good output results (especially for the phoneme posteriors) [9]. The disadvantage of the HMM-based systems lies in the Baum-Welch (BW) training algorithm which is known for his convergence to local maxima. This problem is mostly solved by multiple algorithm initializations which can be extremely time-consuming. Another problem is that these algorithms do not aim on minimization some objective function (e.g. specific loss function) [10].

Few researchers proposed different classifiers with the different results. For example authors in [11] devise a naive Bayes classifier based on the reconstructed phase space. Authors in [5] or [12] proposed new type of feature extraction technique. Most of these approaches do not aim on the improving acoustic models (AM) but rather on defining more sophisticated feature extraction techniques etc. [5], [12]. In the recent years large margin and kernel methods have proven to be an effective tool for the tasks of speech processing (e.g. speech recognition, keyword detection etc.). Most of these systems aim on the acoustic models improving with the use of proper frame-based phoneme classifiers [13], [14]. The frame-based classification is a task of deciding the identity of the each speech frame (typically 25 ms length). Due to the lack of sequence modeling these systems are less accurate compared to the sequence based. The advantage is in the proper application with the specific system, like [10], [14].

Based on the recent advantages in large margin and kernel methods and pioneering research of O. Dekel and J. Keshet [6], [10] this paper presents a simple frame-based linear phoneme classifier. The main idea is in the definition of so called prototype functions for each of the phoneme and the decision is made according to their similarity to each of these functions. Furthermore we have incorporated a distance metric for the hierarchical structure like in Fig. 1. This metric represents a tree induced error over the hierarchical structure and costs misclassified phonemes according to their distance in the tree [6]. For example, classify an utterance as a phoneme *iy* instead *ih* is less severe than phoneme *uw* as a *w*. The hierarchical structure in Fig. 1 represents phonetic tree structure for the American English and a set of the used phonemes is derived from the TIMIT speech corpus [15]. Given the fact that current speech corpora contains large amounts of annotated speech samples we have proposed an efficient learning procedure for the prototype function estimation. This learning algorithm is based on the batch generalization and the results clearly show the benefit of proposed algorithm. Moreover we have proposed a batch-classifier for the whole phoneme sequence classification. The advantage of our approach is that the efficient learning algorithm can be used both for the frame-based classification and whole sequence-based classification.

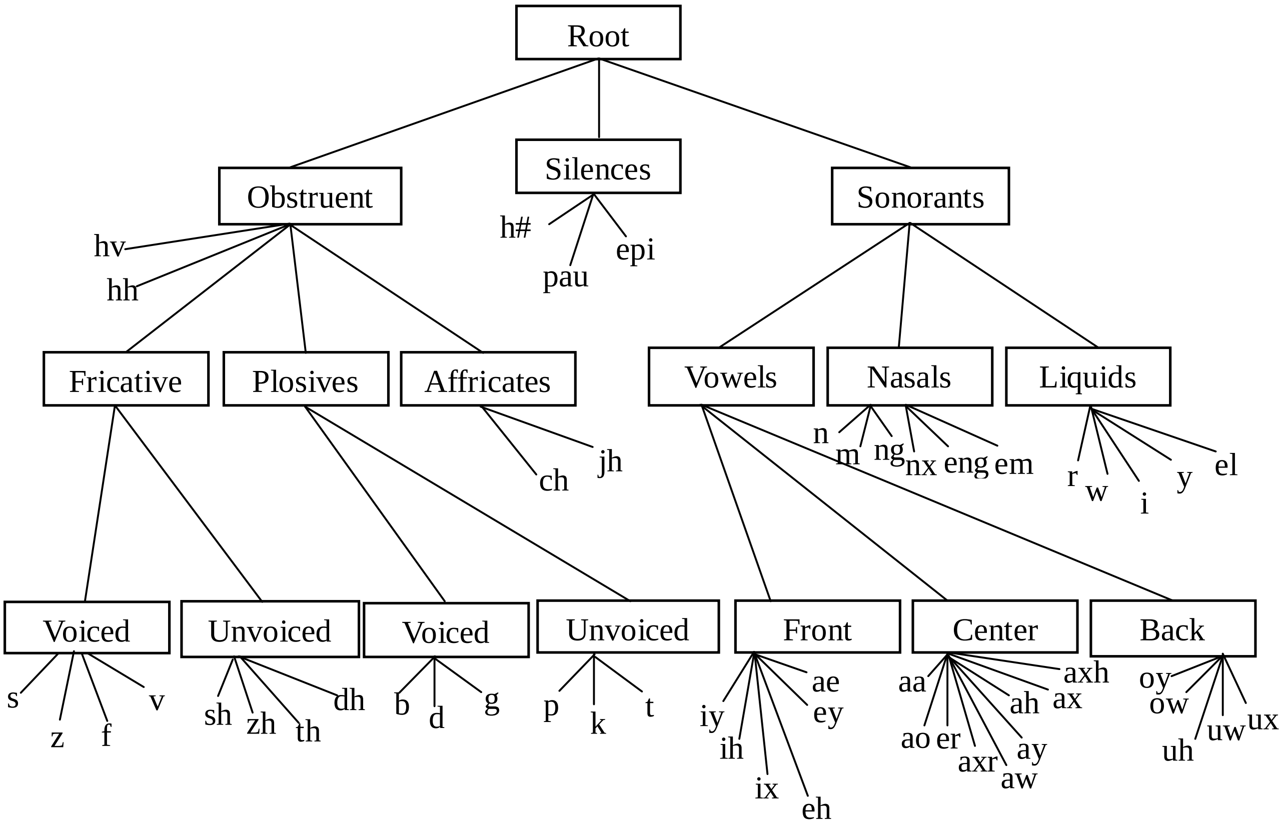

Fig. 1: TIMIT phonetic TREE structure.

For the task of evaluation two different metrics are used. First metric defines a phoneme error rate (PER) which corresponds to the number of misclassified phonemes. Second metric defines number of misclassified phoneme groups (MISS). For example to classify an utterance as *iy* instead of *ih* is a mistake for the PER metric while the metric MISS evaluate correct classification.

This paper is organized as follows. In section 2 we define the problem settings. In section 3 we present a classification rule. Section 4 and 5 introduce our proposed algorithms. In section 6 an evaluation is performed and section 7 concludes our results.

2. Problem Settings

Let $\mathbf{x}$ be the sequence of acoustic feature vector, so $\mathbf{x} = (x_1, x_2 ... x_T)$, $x_T \in \mathbf{X}$, where $\mathbf{X} \subset \mathbf{R}n$ is the acoustic feature domain. Let $\boldsymbol{Y}$ be a set of phonemes and phoneme groups defined according the hierarchical structure like in Fig. 1. Let us further consider align between each of the phoneme or phoneme group $\boldsymbol{y} \in \boldsymbol{Y}$ and appropriate acoustic features $\mathbf{x} \in \boldsymbol{X}$. Denote T to be a corresponding hierarchical structure (like the one in Fig. 1). The number of all vertices in the tree structure T is denoted as a $k = |\boldsymbol{Y}|$, in other words k encompasses the number of all phoneme and phoneme groups so $\boldsymbol{Y} = \{0,..., k-1\}$, where 0 represents a tree root of the hierarchical structure $\boldsymbol{T}$ [6].

Let us define a metric $\gamma(\cdot, \cdot)$ over this hierarchical tree structure $\boldsymbol{T}$ as a number of all edges (unique path) between two different phonemes of phoneme groups u, v. For any pair of phonemes u, v let $\gamma(u, v)$ be their distance in the tree root structure T, while following triangle equality holds $\gamma(u, v) = \gamma(v, u)$ and $\gamma(u, u) = 0$ since γ(u, v) is a non-negative function. based on the stated definitions let us further define a tree induced error $\gamma(u, v)$ as a unique path from the phoneme *u* to *v*, so the tree induced error incurs only while predicting incorrect phoneme [10].

For every phoneme and phoneme group (expect the tree root) $v \in \{Y \backslash 0\}$ we denote $A(v)$ to be a parent of *v* in the *T*. Further we define an ancestor of *v* as a $A^{(i)}(v)$ which is recursively defined as follows,

$$A^{i}(v) = A\left(A^{(i-1)}(v)\right), \qquad (1)$$

and $A^{(0)}(v) = v$. For each phoneme and phoneme group $v \in Y$ we define $P(v)$ to be a number of vertices from phoneme *v* to the tree root 0 resp. $P(v)$ encodes a unique path from phoneme *v* to tree root 0 [6], [10],

$$P(v) = \left\{u \in Y : \exists i, u = A^{(i)}(v)\right\}. \qquad (2)$$

The goal of the frame–based classifier is to determine frame identity – to decide which the most probable phoneme that frame belongs to. In case of the hierarchical based classifier there could be stated another goal. To determine frame’s phoneme group identity. Both of these stated goals can be measured by above mentioned metrics (PER and MISS).

3. Classification Rule

The proposed frame-based classifier (resp. classification function) f: $X \rightarrow Y$ makes its prediction according to the input set of prototypes (weights vectors) W defined for each of the phoneme and phoneme group *v*. Each of the prototype W can be any vector in R*n* and our goal is to train frame-based classifier *f* which attains low tree induced error $\gamma(u, v)$ on the training samples. For the frame-based training algorithm the input training database is defined in the following form *S = {(xi, yi)}m i*=1, where *m* is the number of all training samples (pairs), that is set *S* consists of *m* pairs in the following form $x_i \in X$ and $yi \in Y$ so the training is performed per each of the frame (therefore frame-based). The task of learning is then simplified to find appropriate weights vectors W_1 ... W*k*-1. Linear frame-based classifier *f* is defined according the following formula:

$$f(x) = \underset{v \in Y}{\arg\max}\left(W^{v} \cdot x\right). \qquad (3)$$

The classifier defined by the Eq. (3) does not include hierarchical structure *T*. To incorporate tree root structure *T* into the classification function we have to define a new set of weights vectors w,

$$w^{w} = W^{v} - W^{A^{1}(v)}, \qquad (4)$$

so we had decided to work with the partial differences w rather that with standard weights W. The weight vector W can be furthermore rewritten based on the Eq. (4) as follows,

$$W^{v} = \sum_{u \in P(v)} w^{v}. \qquad (5)$$

Based on the Eq. (5) and (3) the resulting hierarchical frame-based classifier can rewritten into the form of Eq. (6),

$$f(x) = \underset{v \in Y}{\arg\max} \sum_{u \in P(v)} w^{u} \cdot x. \qquad (6)$$

4. Efficient Training Algorithm

The proposed training algorithm is based on the concept of O. Dekel [6] and J. Keshet [10] and both of these algorithms are based on the frame-based classification function (like the one in Eq. (3)) so the learning procedure is also proposed as a frame-based. The principle of our learning algorithm is based on the idea of sequential training and sequence generalization. On each round not a simple frame updates our weights vectors w but rather the whole generalized phoneme frames sequence. In other worlds, on each round all the appropriate weights w of each phoneme *v* are updated at once. Furthermore, our derived prototypes w_v can still be efficient in a frame-based classification as well in whole sequence (batch) classification. The principle of our learning algorithm lies in the redefinition of the classification function *f* defined by the Eq. (6). As stated above, this function is defined to be a frame-based so to incorporate a whole sequence prediction we have to rewrite our classification function as follows,

$$f(x) = \underset{v \in Y}{\arg\max} \sum_{u \in P(v)} mean(w_i^{u} \cdot x), \qquad (7)$$

where operator mean() represents an average of the partial values *xj* , where *j* is a parameter index (e.g. first MFCC coefficient) and w_i is a weight vector in the *i*-th iteration step. In the theory of the Large margin and kernel methods we assume that there exists a set of prototypes $\{w(v)\}v_{\text{in}Y}$ such that for each pair (x_i, y_i) and every $r \neq y_i$ the following inequality holds:

$$\sum_{v \in P(y_i)} \left\| w_i^{v} \cdot x \right\| - \sum_{u \in P(r)} \left\| w_i^{u} \cdot x \right\| \geq \sqrt{\gamma(y_i, r)}, \qquad (8)$$

where y_i is a correct prediction according the classification function defined by Eq. (7) and $\| \cdot \|$ refers to L2 norm. According to the Eq. (8) we require that the difference between the correct prediction and any incorrect prediction is at least square-root of the tree-based distance between them [10]. The goal of the proposed algorithm is to find a set of prototypes which fulfills the margin requirement defined by Eq. (8) while incurring minimal tree-induced error [6]. In machine learning we do not minimize Eq. (8) directly but rather employs a convex hinge-loss function $\ell(\{w_i(v)\}, x_i, y_i)$

$$\ell = \left[\sum_{v\in P(y)} \left\| w_i^v \cdot x \right\| - \sum_{u\in P(y_i)} \left\| w_i^u \cdot x \right\| + \sqrt{\gamma(y_i, y)} \right]_+ , \quad (9)$$

where $[z]+ = \max\{z, 0\}$. Let us assume that there was a prediction mistake b y_i on round i and we would like to modify a set of prototypes $\{w_i(v)\}$ so the constraints defined by Eq. (8) holds. However a simple analytical solution does not exist so we introduce a simple optimization problem frequently used in SVM and machine learning theory [16]. Formally, the new set of prototypes $\{w_{(i+1)}(v)\}$ is the solution of the following optimization problem,

$$\begin{gathered} \min_{\{w^v\}} \frac{1}{2} \sum\nolimits_{v\in Y} \left\| w^v - w_i^v \right\|^2 \\ s.t. \sum_{v\in P(y_i)} \left\| w_i^v \cdot x \right\| - \sum_{u\in P(r)} \left\| w_i^u \cdot x \right\| \geq \sqrt{\gamma(y_i, r)} \end{gathered} . \quad (10)$$

Note, that only the weights $\{w_i(v)\}$ defined by the path $P(v)$ are updated at each iteration. The Fig. 2 demonstrates this update – only the vertices depicted by the solid lines are updated at once.

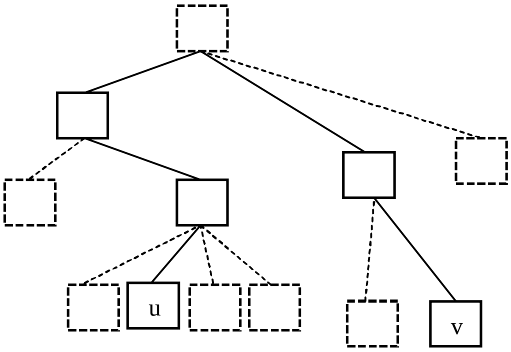

Fig. 2: Re-estimation of the weights vectors – only the solid lines are updated.

The solution to the optimization problem defined by the Eq. (10) is based on the dual representation in the form of Lagrangian [16]. We set the derivate of Lagrangian $\{w(v)\}$ to zero so the new weights $\{w_{(i+1)}(v)\}$ are estimated according the following formulas,

$$w_{i+1}^v = w_i^v + \alpha_i mean(x), \quad v \in P(y_i) \setminus P(\hat{y}_i), \quad (11)$$

$$w_{i+1}^v = w_i^v - \alpha_i mean(x), \quad v \in P(\hat{y}_i) \setminus P(y_i), \quad (12)$$

where the Lagrangian multipliers α_i are simply computed as a

$$\alpha_i = \frac{\ell\left(\left\{w_i^v\right\}, x_i, y_i\right)}{\gamma(y_i, \hat{y}_i) \cdot \left\| x \right\|_N}, \quad (13)$$

where $\|\cdot\|_N$ represents a matrix form defined by the following term

$$\sum_{i=1}^{\max(i)} \sqrt{\sum_{j=1}^{\max(j)} x_{i,j}^2} . \quad (14)$$

On each round we have generated a new set of prototypes $\{wv\ i\}$ so while the training corpus S contains m training samples we have m sets of prototypes. The last set of weight vectors $\{w_m(v)\}$ should be the best resulting prototypes but in practice an averaged weights vectors shows to be more efficient. The resulting prototypes are defined as follows,

$$w_{avg}^v = \frac{1}{m+1} \sum_{i=1}^{m+1} w_i^v . \quad (15)$$

```
INITIALISATION: $\forall v \in Y$: wv 1 = 0
For i=1, 2. . . m
    – Algorithm receive acoustic features vector _xi for the phoneme yi
    – Prediction
          $f(x) = \arg\max_{v\in Y} \sum_{u\in P(v)} mean(w_i^u \cdot x)$
    – Correct phoneme yi is revealed
    – In case of incorrect prediction ($\gamma(\cdot,\cdot) \neq 0$) the hinge loss function $\ell(\{w_i(v)\}, x_i, y_i)$ is computed
    – Re-estimation of the weight vectors:
          $w_{i+1}^v = w_i^v + \alpha_i^v mean(x)$
          $\alpha_i^v = \begin{cases} \alpha_i & v \in P(y_i) \setminus P(\hat{y}_i) \\ -\alpha_i & v \in P(\hat{y}_i) \setminus P(y_i) \\ 0 & otherwise \end{cases}$
    – Where
          $\alpha_i = \frac{\ell\left(\left\{w_i^v\right\}, x_i, y_i\right)}{\gamma(y_i, \hat{y}_i) \cdot \left\| x \right\|_N}$
```

Fig. 3: Proposed training algorithm.

5. Non-Linear Classifiers

The proposed training algorithm can be further incorporated with the non-linear kernel transformation. The main idea lies in the vector space separation.

Because of speech complexity not all of the frames can be linearly well separated in the input feature space R*n*. Based on the SVM theory there a non-linear transformation can be applied on the input features (both in the training and evaluation) [16]. Again, transformed features are linearly separated (non-linearly decision hyper plane can be seen in the original feature space).

To define a non-linear classifier we have to rewrite our fundamental classification rule defined by the Eq. (3) in case of the linear classifier and Eq. (6) in case of the hierarchical classifier. Because this paper primary deals with the hierarchical classifiers Eq. (6) will be rewritten but the same can be applied on the linear classifiers. The resulting classifier will be in the following form:

$$f(x) = \arg\max_{v\in Y} \sum_{u\in P(v)} \sum_{i=1}^{m} \alpha_i \cdot x_i \cdot x\,, \qquad (16)$$

where α_i is *i*-th Lagrangian multiplier and *xi* is *i*-th training sample. This definition is valid and expressing the whole weight vector w estimation. Equation (16) can be further rewritten in the Kernel notation in the form of following Eq. (17)

$$f(x) = \arg\max_{v\in Y} \sum_{u\in P(v)} \sum_{i=1}^{m} \alpha_i \cdot K(x_i \cdot x)\,, \qquad (17)$$

where inner product between x_i and x is represented by the kernel operator *K*. Based on the Eq. (16) and (17) there should be clear that the whole training database (or at least Lagrangians α_i) are necessary in the classification process. Nevertheless, only non-negative Lagrangians contributes to the final result which leads to the sparse solution.

$$\ell = \left[\sum_{v\in P(\hat{y}_i)} \sum_{j<i} \left\| \alpha_j^v \cdot K(x_i, x_j) \right\| - \sum_{u\in P(y_i)} \left\| \alpha_j^v \cdot K(x_i, x_j) \right\| + \gamma(y_i, y) \right]_+ . \qquad (18)$$

5.1. Training Algorithm

The advantage of the re-definition according to Eq. (17) lies in the possibility of kernel *K* pre-computation. Kernel *K*(x$_i$, x) can be reformulated as *K*(x$_i$, x$_j$), where x$_i$ and x$_j$ are training samples where *i*, *j* = 1 . . .*m*. Resulting matrix G (so called Gram matrix [16]) is composed with every *K*(x$_i$, x$_j$) value and this matrix can be pre-computed just once (for the same kernel parameters).

To develop a training algorithm we have further incorporate kernel operator into the loss function ℓ. This leads to the Eq. (18), where Lagrangians α$_i$*(v)* are computed based on the following equation

$$\alpha_i = \frac{\ell\left(\left\{\alpha_j^v\right\}, G(j,i), y_i\right)}{\gamma(y_i, \hat{y}_i) \cdot G(i,i)}\,, \qquad (19)$$

where $\alpha_j(v)$ is *j*-th Lagrangian associated with the phoneme y_j and G represents Gram matrix.

We have further experimented with the mutual combination of linear and non-linear classifiers which lead to the following efficient training algorithm – see Fig. 5.

INITIALISATION: $\forall v \in Y$: w$_1$*(v)* = 0, $\alpha_i(v) = 0$

Pre-computation of Gram matrix **G** [optional]

$$G(i, j) = K(x_i, x_j)$$

For i=1, 2. . . m

- Algorithm receive acoustic features vector x_i for the phoneme y_i
- Prediction

$$f(x) = \arg\max_{v\in Y} \sum_{u\in P(v)} mean(w_i^u \cdot x)$$

- Correct phoneme y$_i$ is revealed
- In case of incorrect prediction ($\gamma(\cdot, \cdot) \neq 0$) the hinge loss function *ℓ({α},G(j,i),yi)* is computed
- Re-estimation of the weight vectors:

$$w_{i+1}^v = w_i^v + \alpha_i^v mean(x)$$

$$\alpha_i^v = \begin{cases} \alpha_i & v \in P(y_i) \setminus P(\hat{y}_i) \\ -\alpha_i & v \in P(\hat{y}_i) \setminus P(y_i) \\ 0 & otherwise \end{cases}$$

- Where

$$\alpha_i = \frac{\ell\left(\left\{\alpha_j^v\right\}, G(j,i), y_i\right)}{\gamma(y_i, \hat{y}_i) \cdot G(i,i)}$$

Fig. 4: Efficient non-linear training algorithm.

6. Evaluation

We have performed a number of tests to evaluate our proposed training algorithm and all algorithms were evaluated over the TIMIT speech corpus [15] which is a speech corpus of annotated utterances for American English. We have divided the TIMIT sentences into the two disjoints groups – TRAIN and TEST. We have also excluded all the *SA* sentences (dialect sentences) and we have randomly generated 80 TRAIN (for the second experiment, number of training examples will vary) and 80 TEST sentences as follow – Each sentences is uttered by the different speaker, each speaker uttered one *SI* and *SX* sentence and both sets have a uniformly distributed all the dialect regions. We have also separated all training and testing sentences so performed evaluation can be considered as a speaker independent.

In the first experiment, classifiers were compared on the two different feature extraction techniques - mel-frequency cepstral coefficients (MFCC) and perceptual linear prediction coefficients (PLP) both detailed described in the literature [17]. We used 13 basic coefficients, deltas and double deltas (Δ+ΔΔ). We have used 15 mixtures for GMM model. Furthermore, features were normalized using the CMN/CVN technique. The Tab. 1 and 2 displays the results indicating the advantage of PLP features. The first proposed training algorithm (denoted as a Hier$_{sekv}$) had been evaluated as a standard linear classifier and shown to be more accurate compared to the frame-bases training algorithm based on the [6]. Moreover, learning time was rapidly reduced. For notation, both training algorithms were evaluated based on the same frame-based classification rule defined by the Eq. (6). The second proposed training algorithm (denoted as a Hier_kernel$_{sekv}$) incorporated a non-linear transformation represented by the kernel operator *K*. To evaluate benefits of our proposed training algorithm we had compared our nonlinear training algorithm with the non-linear training algorithm proposed in the [6] (denoted as a Hier_kernel). Finally, all hierarchical frame-based training algorithms were compared with the standard GMM frame-based classifier (like the one in [4], [7]). To assure a convergence to global optimum we have performed a number of re-estimation of the training algorithm and the one yielding the best results over the cross-validation set had been chosen for the further evaluation.

Tab.1: PER and MISS for PLP features.

Number of sentences [-]	**PER [%]**	**MISS [%]**	**Training time [min]**
Hier	55	31	180
Hier$_{sekv}$	53	29	**15,8**
Hier_kernel	54	29	315
Hier_kernel$_{sekv}$	**49**	**25**	195
GMM	52	36	35

Our second experiment demonstrates a benefit of our training algorithm. Table 3 shows that with the larger number of training sentences the proposed sequence based algorithm converges to the global optimum defined by the frame-based learning algorithm. Furthermore, at the same PER and MISS results our proposed algorithm is much more time-efficient compared to the frame-based algorithm based on the [6]. Figure 3 and 4 graphically output these results.

Tab.2: PER and MISS for MFCC features.

Classifier type	**PER [%]**	**MISS [%]**	**Training time [min]**
Hier	56	32	182
Hier$_{sekv}$	55	31	**16,7**
Hier_kernel	54	29	315
Hier_kernel$_{sekv}$	**50**	**26**	196
GMM	53	36	36

Tab.3: PER and MISS for different number of training sentences (for PLP features).

Number of sentences [-]	**PER [%]**	**MISS [%]**	**Training time [min]**
80	64	40	6,8
160	59	36	10
240	53	31	15,8

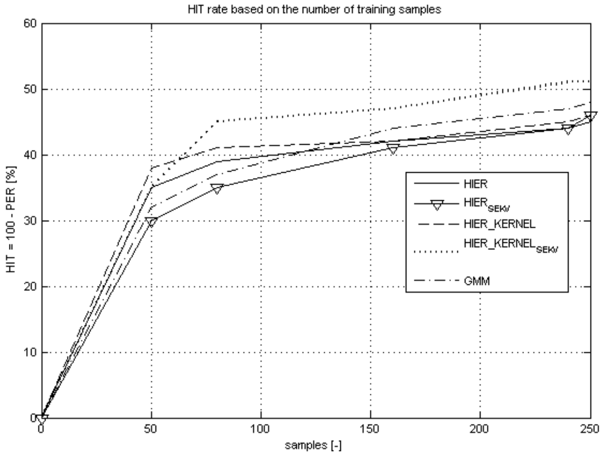

Fig. 5: Classifier precision (HIT) based on the number of input training samples.

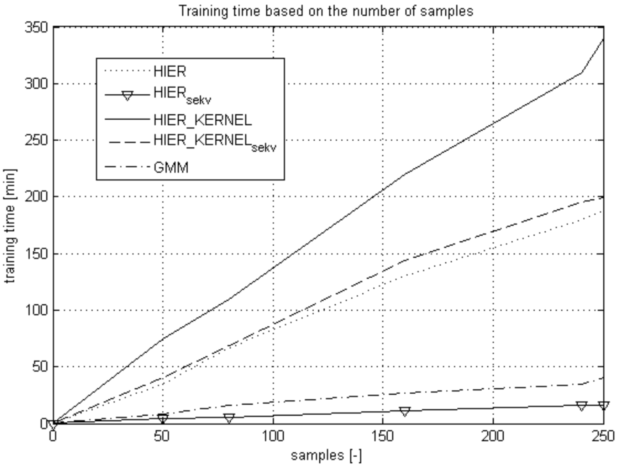

Fig. 6: Overall classifier training time.

7. Conclusion

This paper compared two state-of-art approaches to frame-based phoneme classification – generic and discriminative. Two state-of-art discriminative frame-based classifiers were presented (denoted as HIER and HIER_KERNEL) along with state-of-art generic classifier represented by the GMM frame-based classifier (denoted as GMM). Moreover, this paper had proposed two efficient training algorithms for discriminative frame-based phoneme classification (denoted with the SEKV suffix). For notation, all discriminative classifiers exploit a hierarchical tree root structure which is inducing tree root metric over the input group of phonemes. Both HIER and HIER_KERNEL classifiers had similar results on PER compared to the GMM classifier, but the results for metric MISS show the advantage of these classifiers (especially the implementation of hierarchical structure had shown to be a very effective). Both proposed training algorithms clearly outperforms all of the previous classifiers and showed possible future direction for frame-based phoneme classification. The results also showed superiority of the PLP features over the MFCC features.

Our future work will be focused on the implementation hierarchical tree root structure into the GMM classifiers and incorporation of long temporal content into the frame-based classifiers. The future effort will also aim on the classifiers evaluation within the KWS systems.

References

[1] PFEIFER, V.; BALIK, M.; MALY, J. Frame based phoneme classification using large margin and kernel methods. In *33rd International Conference on Telecommunications and Signal Processing, TSP 2010*, Baden, Austria, September 2010, p. 1-4. ISBN 978-963-88981-0-4.

[2] PFEIFER, V.; BALIK, M. Spotting techniques used in modern videoconference systems. In *31st International Conference on Telecommunications and Signal Processing, TSP 2008*, September 2008, pp. 55–58. ISBN 978-963-06-5487- 6.

[3] PFEIFER, V.; BALIK, M. Effective plagiarism detection system for advanced videoconference systems. In *32nd International Conference on Telecommunications and Signal Processing, TSP 2009*, Asszisztencia Szevezo kit. H-1136 Budapest. Hedegus Gyula u. 20, September 2009, pp. 101–106. ISBN 978-963-06-7716- 5.

[4] PFEIFER, V.; BALIK, M.; MICA, I.; PRUSA, Z. Phoneme confidence estimator based on gaussian mixture models. *Gests Internation Transactions on Communication and Signal Processing*, 2009. Vol. 13, No. 13, p. 1-8, ISSN 1738-9682.

[5] RFC 3611. *RTP* Control *Protocol Extended Reports*. Paris: The Internet Society, 2003. pp. 54

[6] RUSEK, Krzysztof; ORZECHOWSKI, Tomasz; DZIECH, Andrzej. LDA for Face Profile Detection. In *Communications in Computer and Information Science: Proceedings 4th International Conference Multimedia Communications, Services and Security Multimedia Communications, Services and Security 2011*. 1st ed. Berlin: Springer, 2011. pp. 144-148. ISBN 978-3-642-21511-7.

[7] DEKEL, O.; KESHET, J.; SINGEER, Y. *An online algorithm for hierarchical phoneme classification*. Springer, vol. 3361, 2005, pp. 146–158, January 2005. ISBN 3-540-24509-X.

[8] PFEIFER, V.; BALIK, M. Fonemovy klasikator zalozen na pravdepodobnostnich modelech. *Elektrorevue*, Vol. 64, p. 1-6, Prosinec 2009. Available at WWW: <http://www.elektrorevue.cz/cz/download/ramcova-hiearchicka-klasi-kace-fonem--ceskeho-jazyka>. ISSN 1213-1539.

[9] GREZL, F. *Trap-based probabilistic features for automatic speech recognition*. Brno, 2007. Ph.D. dissertation, Brno University of Technology.

[10] ARADILLA, G.; VEPA, J.; BOURLARD, H. *Using posterior-based features in template matching for speech recognition*. IDIAP research institute, Tech. Rep. IDIAP-PR 06-23, June 2006.

[11] KESHET, J. *Large margin algorithms for discriminative continuous speech recognition*. Hebrew, 2007. Ph.D. dissertation, Hebrew University.

[12] YE, J.; POVINELLI, R. J.; JOHNSON, M. T. Phoneme classification using naive bayes classifier in reconstructed phase space. In *Digital Signal Processing Workshop*, 2002 and the *2nd Signal Processing Education Workshop*. Proceedings of IEEE 10th, 2002. pp. 37–40. ISBN 0-7803-8116-5.

[13] GRANGIER, D.; BENGIO, S. *A discriminative decoder for the recognition of phoneme sequences*. IDIAP Research Institution, Tech. Rep., November 2005.

[14] KESHET, J.; BENGIO, S. Eds. *Automatic Speech and Speaker Recognition: Large Margin and Kernel Methods*. Wiley, January 2009, vol. 1. pp. 268. ISBN 978-0470696835.

[15] KESHET, J.; GRANGIER, D.; BENGIO, S. Discriminative keyword spotting. In *Workshop on Non-Linear Speech Processing (NOLISP)*, 2007, p. 1-5. [Online]. Available at WWW: <http://david.grangier.info/pub/papers/2007/KeshetGrBe07.pdf>.

[16] GAROFOLO, J. S.; LAMEL, L. F.; FISHER, W. M.; FISCUS, J. G.; PALLETT, D. S.; DAHLGREN, N. L.; ZUE, V. Timit acoustic-phonetic continuous speech corpus. In *10th International Conference on Speech and Computer. Linguistic Data Consortium*, 1993, p. 4.

[17] BISHOP, C. M. *Pattern recognition and Machine learning*, B. S. M. Jordan, J. Kleinberg, Ed. Springer, February 2006, Vol. 1. pp. 740. ISBN 978-0-387-31073-2.

[18] PSUTKA, J.; MULLER, L.; MATOUSEK, J.; RADOVA, V. *Mluvíme s počítačem česky*. ACADEMIA, October 2006. pp. 746. ISBN 80-200-1309-1.

[19] HERMANSKY, H.; ELLIS, D. P.; SHARMA, S. Tandem connectionist feature extraction for conventional hmm systems. In: *Proceedings of the conference ICASSP-2000*, 2000, p. 1-4. [Online]. Available at WWW: <ftp://ftp.icsi.berkeley.edu/ pub/speech/papers/icassp00-nnhmm.pdf>. ISBN 0-7803-6293-4.

Vaclav PFEIFER was born in 1982. He received his M.Sc. from electronic and communications in 2006. His research interests include speech processing focused on the spoken term detection and reconstruction of the object trajectory based on the two rig cameras. He is author and co-author of many articles dealing with the spoken term detection and currently he is finalizing his thesis to complete Ph.D. studies.

Miroslav BALIK was born in 1973. He received the Ph.D. degree at the Faculty of Electrical Engineering and Communication of Brno University of Technology in 2003. In present time he focuses on noise reduction algorithms based on VOLA segmentation method. His research interests include “Digital Audio Signal Processing”, “Multimedia Systems and Services” and “Computer Systems”. He is the author of several digital systems for musical signal real-time processing. He is a senior member of IEEE.