An SVM Plait for Improving Affect Recognition in Intelligent Tutoring Systems

Ruth Janning, Carlotta Schatten and Lars Schmidt-Thieme
Information Systems and Machine Learning Lab (ISMLL)
University of Hildesheim, Germany
Email: janning@ismll.uni-hildesheim.de

Gerhard Backfried and Norbert Pfannerer
SAIL LABS Technology AG
Vienna, Austria
Email: Gerhard.Backfried@sail-labs.com

***Abstract*—Usually, in intelligent tutoring systems the task sequencing is done by means of expert and domain knowledge. In a former work we presented a new efficient task sequencer without using the expensive expert and domain knowledge. This task sequencer only uses former performances and decides about the next task according to Vygotsky's Zone of Proximal Development, that is to neither bore nor frustrate the student. We aim to support this task sequencer by a further automatically to gain information, namely students affect recognized from his speech input. However, the collection of the data from children needed for training an affect recognizer in this field is challenging as it is costly and complex and one has to consider privacy issues carefully. These problems lead to small data sets and limited performances of classification methods. Hence, in this work we propose an approach for improving the affect recognition in intelligent tutoring systems, which uses a special structure of several support vector machines with different input feature vectors. Furthermore, we propose a new kind of features for this problem. Different experiments with two real data sets show, that our approach is able to improve the classification performance on average by 49% in comparison to using a single classifier.**

***Keywords*-intelligent tutoring systems; affect recognition; support vector machine (SVM); speech features; affect recognition performance improvement; plait structure;**

I. Introduction

Nowadays, intelligent tutoring systems are an important tool for supporting the education of students for instance in learning mathematics. The main advantages of intelligent tutoring systems are the possibility for a student to practice any time and anywhere, as well as the possibility of adaptivity and individualization for a single student. Usually, an adaptive intelligent tutoring system possesses an internal model of the student and a task sequencer which decides which tasks in which order are shown to the student. Originally, the task sequencing in adaptive intelligent tutoring systems is done using information gained from expert and domain knowledge and logged information about the performance of students in former exercises. In [19] a new efficient sequencer based on a performance prediction system was presented, which only uses former performance information from students to sequence the tasks and does not need the expensive expert and domain knowledge. This approach applies for performance prediction the machine learning method matrix factorization (see e.g. [4]) to former performance information. Subsequently, it uses the output of the performance prediction process to sequence the tasks according to the theory of Vygotsky's Zone of Proximal Development [22]. That is the sequencer chooses the next task in order to neither bore nor frustrate the student or in other words, the next task should not be too easy or too hard for the student. In this work we aim to support this kind of task sequencing in intelligent tutoring systems by affect recognition applied to speech input from the students interacting with the system while solving tasks. Appropriate to the used theory of Vygotsky's Zone of Proximal Development we try to classify features gained from the speech input as 'the student was *over-challenged* by the last task', 'the student was *under-challenged* by the last task' or 'the student was in a *flow*'. This information can be used to decide about the next task. At first glance it seems to make sense to use as features words related to affects after a speech recognition was applied. However, this approach is dependent on the performance of the speech recognition and inherits its error. Hence, we decided to use features extracted directly from the sound files, like features gained from speech pauses, or features gained from initial processing steps of speech recognition. The most efficient state-of-the-art classification approach for the kind of features used is a support vector machine (SVM, see e.g. [20] and [16]). However, usually the collection of data from young students is costly and complex and one has to consider privacy issues carefully. These facts lead to small data sets in this area and finally to a limited performance of the support vector machine applied to the data. Hence, the question arises, if there is a way to improve the classification performance. A hybrid neural network plait (HNNP) for improving the classification performance of artificial neural networks applied to signal data like for instance Ground Penetrating Radar data (see [9]) or phonemes (see [1], [14]) was presented in [10]. The idea of this paper is to adapt the plait principle of the HNNP approach to a structure of support vector machines applied to features from speech data. The main contributions of this paper are: (1) proposal and investigation of new features for affect recognition in intelligent tutoring systems, (2) proposal of an SVM plait structure for improving the affect recognition performance and (3) different experiments with real data proving the effectiveness of the proposed features and SVM plait. In the following, we will present after the related work section II the proposed features in section III and the proposed SVM plait structure in section IV. The different experiments and their results are presented and discussed in section V.

II. Related Work

Support vector machines (SVM, [2], [5]) are supervised machine learning methods which can be used for classification tasks and deliver in many areas the best performance in comparison to other classification approaches. The library LIBSVM ([3]) delivers an efficient and often used implementation of an SVM. In the field of emotion and affect recognition SVMs are state-of-the-art for features extracted for instance from speech data (see e.g. [20], [16]). Those features can be disfluencies features like the ones used for expert identification in [23], [18] and [15], or for emotion recognition in [17]. In this work we investigate two different own kind of features: amplitude and articulation features (see section III). The amplitude features were already proposed in former work (see [12], [13]), whereas the proposed articulation features are new work.

The idea for our proposed SVM plait approach is based on former work ([10], [11], [14]), where we developed and investigated a hybrid neural network plait (HNNP) for improving the classification performance on small and noisy signal data sets. The HNNP approach uses different feature sets from different information sources and different kinds of neural networks with adapted architecture which are retrained interactively within a plait structure using additional side information gained before and during the retraining for a further improvement. The SVM plait has a similar structure but it uses the same kind of SVMs within the plait structure and feature subsets of one information source. The architecture of the SVMs within the SVM plait structure do not need to be adapted, the additional input is added instead to the input feature vectors. The proposed plait structure uses the principles of ensemble methods like stacking, or stacked generalization respectively, which are explained and investigated for instance in [21]. Different to plain stacking the SVM plait consists of several layered stackings with different additional inputs. A cascade of SVMs is presented in [6], but the goal of that approach is to enable a parallelization of SVMs for computing large data.

III. Speech Features and Affect Classes

As mentioned above for our approach we use features extracted from speech data. We propose two different kinds of those features – features from amplitudes and articulation features – which are described in the following two subsections. The class labels to which the features shall be mapped come from the theory of Vygotsky's Zone of Proximal Development and can be summarized as *perceived task-difficulty labels*: *over-challenged*, *flow* and *under-challenged*. For the sake of simplicity and as examples with label *under-challenged* can be observed rarely (in our real data sets there were only 2 of those examples), in this work we focus on the binary classification problem of distinguishing between examples of class *over-challenged* and class *flow*. With more data and more examples of class *under-challenged* one has simply to adapt the presented approach to a multi-class problem by substituting the originally used support vector machines by multi-class support vector machines (see also section IV).

A. Features from Amplitudes

The first kind of features (see also [12], [13]) are features gained from the amplitudes, or the decibel values respectively, of the sound files. More explicitly, the decibel values are used to identify pauses within the speech input data. This is done by defining a threshold on the decibel scale (as done e.g. in [15]) which designates which decibel values belong to speech and which ones to pauses (see figure 1). We adjusted the threshold for our experiments in section V by hand, but later on – for the application phase – the threshold should be learned.

The advantage of using features gained from amplitudes is that instead of a full speech recognition approach only a pause identification by means of the mentioned threshold has to be applied before computing the features. That deceases the complexity of the affect recognition and one does not inherit the error of the full speech recognition approach, which makes the system more noise robust. Furthermore, these features are independent from the need that students use words related to affects.

To compute features for a task i presented to a student first we extract some measurements from the sound file of the task, or from pause information gained from the decibel values respectively:

- the total length of pauses p_i and the total length of speech s_i in the sound file,
- the number of pause segments n_{p_i} and the number of speech segments n_{s_i} within the speech input,
- the uth pause segment $p_i^{(u)}$ and the wth speech segment $s_i^{(w)}$ within the speech input,
- the seconds t_i needed by the student to solve the task.

The final features $x_i^0, \ldots, x_i^7$ built from the extracted measurements for a task i are the following:

$$x_i^0 = \frac{p_i}{s_i} \tag{1}$$

(Ratio between pauses and speech)

$$x_i^1 = n_{p_i} + n_{s_i} \tag{2}$$

(Frequency of speech pause changes)

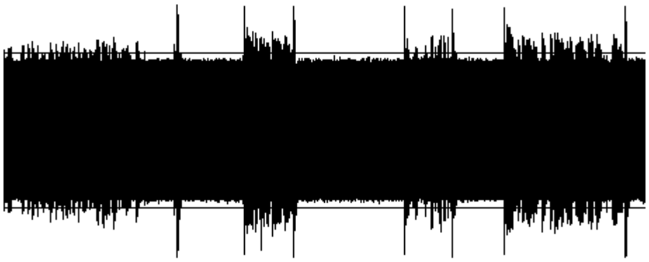

Figure 1. Graphic of the decibel scale of an example sound file of a student. The two straight horizontal lines indicate the threshold.

$$x_i^2 = \frac{p_i}{(p_i + s_i)} \tag{3}$$

(Percentage of pauses of input speech data)

$$x_i^3 = \max_u(p_i^{(u)}) \tag{4}$$

(Length of maximal pause segment)

$$x_i^4 = \frac{\sum_u p_i^{(u)}}{n_{p_i}} \tag{5}$$

(Length of average pause segment)

$$x_i^5 = \max_w(s_i^{(w)}) \tag{6}$$

(Length of maximal speech segment)

$$x_i^6 = \frac{\sum_w s_i^{(w)}}{n_{s_i}} \tag{7}$$

(Length of average speech segment)

$$x_i^7 = t_i \tag{8}$$

(Seconds needed for the task)

The output of this feature extraction process is a feature vector $\mathbf{x}_i = (x_i^0, \ldots, x_i^7), i = 1, \ldots m$, where m is the number of examples and the appropriate class label is y_i.

The idea behind this kind of features came from the observation that often children exhibit longer pauses of silence while thinking about the problem when they are *over-challenged* or talk with less and smaller pauses when they are in a *flow* (see also [13]).

B. Articulation Features

The second kind of new features we propose are articulation features gained from an intermediate step of the speech recognition process. In preparation for the speech recognition in this step the speech input is partitioned into segments consisting of vowels, consonants – obstruents or fricatives – and silence tags (as well as some non-speech features like breathing, but those occurred rarely in our collected real data, hence we did not use them for this work). We use this preliminary information to create a new kind of features for a task i. To compute these features first we have to extract some measurements:

- the number of silence tags n_{sil_i}, the number of vowels n_{V_i}, the number of obstruents n_{O_i} and the number of fricatives n_{F_i} within the speech input of task i,
- the lengths $V_i{}^{(u)}$, $O_i{}^{(w)}$, $F_i{}^{(z)}$ and $sil_i{}^{(r)}$ of the uth vowel, wth obstruent, zth fricative and the rth silence tag within the speech input of task i.

The final features $x_i^0, \ldots, x_i^{12}$ gained from the extracted measurements are the following:

$$x_i^0 = n_{sil_i} \tag{9}$$

(Number of silence tags)

$$x_i^1 = \frac{\sum_u V_i{}^{(u)}}{n_{V_i}} \tag{10}$$

(Average length of vowels)

$$x_i^2 = \frac{\sum_w O_i{}^{(w)}}{n_{O_i}} \tag{11}$$

(Average length of obstruents)

$$x_i^3 = \frac{\sum_z F_i{}^{(z)}}{n_{F_i}} \tag{12}$$

(Average length of fricatives)

$$x_i^4 = \frac{\sum_r sil_i{}^{(r)}}{n_{sil_i}} \tag{13}$$

(Average length of silence tags)

$$x_i^5 = \max_u(V_i^{(u)}) \tag{14}$$

(Maximal length of vowels)

$$x_i^6 = \max_w(O_i^{(w)}) \tag{15}$$

(Maximal length of obstruents)

$$x_i^7 = \max_z(F_i^{(z)}) \tag{16}$$

(Maximal length of fricatives)

$$x_i^8 = \max_r(sil_i{}^{(r)}) \tag{17}$$

(Maximal length of silence tags)

$$x_i^9 = \min_u(V_i^{(u)}) \tag{18}$$

(Minimal length of vowels)

$$x_i^{10} = \min_w(O_i^{(w)}) \tag{19}$$

(Minimal length of obstruents)

$$x_i^{11} = \min_z(F_i^{(z)}) \tag{20}$$

(Minimal length of fricatives)

$$x_i^{12} = \min_r(sil_i{}^{(r)}) \tag{21}$$

(Minimal length of silence tags)

The output of this feature extraction process is a feature vector $\mathbf{x}_i = (x_i^0, \ldots, x_i^{12}), i = 1, \ldots m$, where again m is the number of examples and the appropriate class label is y_i.

The idea behind this kind of features is that depending on the affective state the person speaking lengthens or shortens vowels, obstruents or fricatives.

IV. SVM Plait

The state-of-the-art method for classification applied to the kind of features described above in section III is a support vector machine (SVM). An SVM (see [2], [5]) is a classifier which searches for a hyperplane which optimally – with maximal margin – separates the examples of different classes in the space of the example vectors. By means of the kernel trick and a kernel function also non-linear problems can be solved by an SVM. Originally, an SVM solves binary classification problems but can be extended to multi-class classification problems (see [3]). As mentioned above we consider a binary classification problem in this paper. For finding the mentioned optimal hyperplane the following optimization problem has to be solved: minimize for $\mathbf{w}, b, \xi$

$$\frac{1}{2} \left|\left|\mathbf{w}\right|\right|_2^2 + C \sum_{i=1}^{m} \xi_i \tag{22}$$

subject to $y_i \left(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b \right) \geq 1 - \xi_i$, $\xi_i \geq 0$, for all $1 \leq i \leq m$. In the formula above $\mathbf{w}$ is a normal vector, b is a bias, C is a constant, m is the number of examples which consist of a feature vector $\mathbf{x}_i$ and the class label $y_i, i = 1, \ldots, m$, the $\xi_i \geq 0$ are slack variables and ϕ is a function mapping the data to a higher dimension to apply the kernel trick. Usually, this optimization problem is solved in its dual form: maximize for α

$$\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j k \left(\mathbf{x}_i \mathbf{x}_j \right) \tag{23}$$

subject to $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^{m} \alpha_i y_i = 0$. The appropriate classification rule is:

$$\begin{aligned} f(\mathbf{x}) &= \operatorname{sgn} \left(\langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b \right) \\ &= \operatorname{sgn} \left(\sum_{i=1}^{m} \alpha_i y_i k \left(\mathbf{x}_i, \mathbf{x} \right) + b \right) . \end{aligned} \tag{24}$$

In formula (23) and (24) the α_i are Lagrange multipliers, k is the kernel function ($k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$), C is a constant appearing as an additional constraint on the Lagrange multipliers, sgn is the sign function, and $\mathbf{w}$ is the normal vector ($\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \phi(\mathbf{x}_i)$).

For our proposed approach the described single SVMs are interweaved within a plait structure (see figure 2) by combining the classification decisions of SVMs in former plait layers with the feature vectors and feeding this combined new feature vectors into further SVMs. In this way the classification performance is improved over the plait layers, as SVMs in later layers learn how to consider the classification decisions of previous SVMs to improve their own classification performance. The feature vectors for the SVMs $\text{SVM}_1^{(0)}$, $\text{SVM}_2^{(0)}$, $\text{SVM}_3^{(0)}$ in the first plait

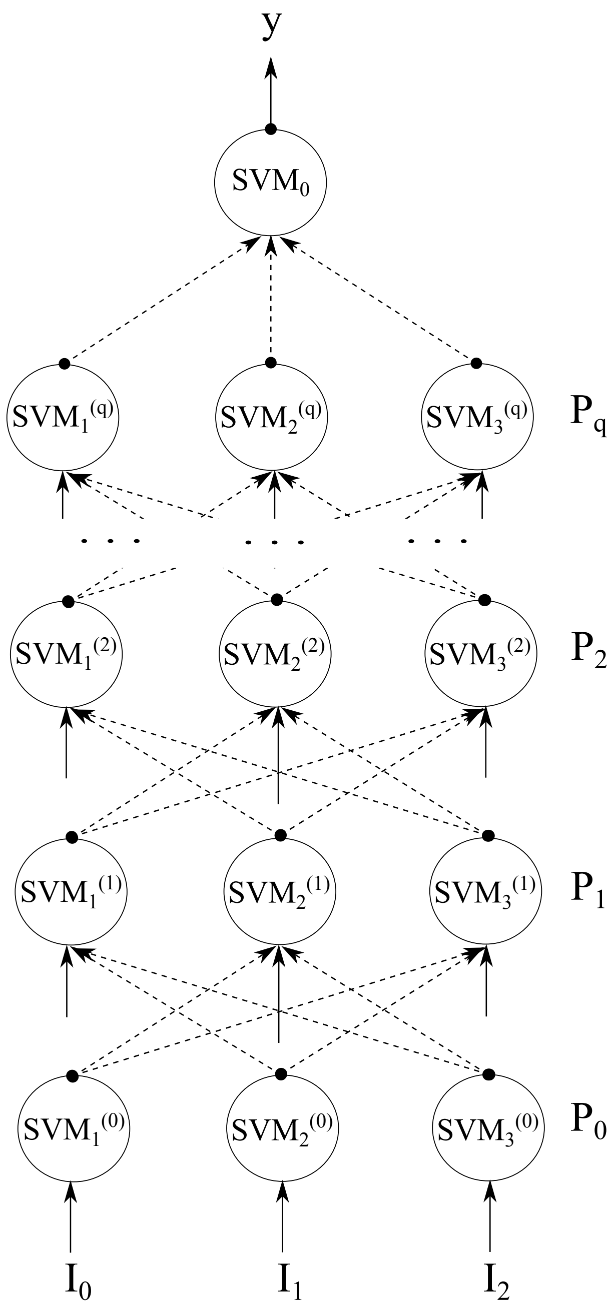

Figure 2. Architecture of the SVM plait. The plait is composed of $q+1$ layers $P_0, P_1, \ldots, P_q$ (q is a hyper parameter). Each layer contains 3 SVMs, which get different feature vectors as input. In every plait layer from P_1 on the SVMs are retrained with enhanced input feature vectors. The enhancement is information from the former layer, namely the outputs (the predicted class labels) of the SVMs in the previous plait layer. After the last plait layer P_q a further SVM (SVM_0) is attached to achieve one common output y delivering the final classification result.

layer P_0 stem only from the original feature vector $\mathbf{x}_i = (x_i^0, \ldots, x_i^l), i = 1, \ldots, m$, $(l+1) =$ number of features, with amplitude ($l = 7$) or articulation ($l = 12$) features as described in section III. The original feature vector $\mathbf{x}_i$ is divided into as many vectors as there are SVMs in one plait layer, i.e. in figure 2 the first input vectors are:

$$I_0 = \mathbf{x}_i^{\mathrm{SVM}_1^{(0)}} = (x_i^0, \ldots, x_i^{(1\cdot\frac{l}{3})}) \; , \tag{25}$$

$$I_1 = \mathbf{x}_i^{\mathrm{SVM}_2^{(0)}} = (x_i^{(1\cdot\frac{l}{3})+1}, \ldots, x_i^{(2\cdot\frac{l}{3})}) \; , \tag{26}$$

$$I_2 = \mathbf{x}_i^{\mathrm{SVM}_3^{(0)}} = (x_i^{(2\cdot\frac{l}{3})+1}, \ldots, x_i^l) \; . \tag{27}$$

If l is too small (like for the amplitude features) then the input vectors I_0, I_1 and I_2 also may overlap to ensure that there are enough feature values within one vector for a good classification performance of the single SVMs. The input feature vectors for the later layers within the plait structure are different. That is the input feature vectors $\mathbf{x}_i^{\mathrm{SVM}_1^{(d)}}$, $\mathbf{x}_i^{\mathrm{SVM}_2^{(d)}}$, $\mathbf{x}_i^{\mathrm{SVM}_3^{(d)}}$ for the SVMs $\mathrm{SVM}_1^{(d)}$, $\mathrm{SVM}_2^{(d)}$, $\mathrm{SVM}_3^{(d)}$ in plait layer P_d are enhanced by two further inputs:

$$\begin{aligned} \mathbf{x}_i^{\mathrm{SVM}_1^{(d)}} &= (I_0, \hat{y}_i^{\mathrm{SVM}_2^{(d-1)}}, \hat{y}_i^{\mathrm{SVM}_3^{(d-1)}}) \\ &= (\mathbf{x}_i^{\mathrm{SVM}_1^{(0)}}, \hat{y}_i^{\mathrm{SVM}_2^{(d-1)}}, \hat{y}_i^{\mathrm{SVM}_3^{(d-1)}}) \\ &= (x_i^0, \ldots, x_i^{(1\cdot\frac{l}{3})}, \hat{y}_i^{\mathrm{SVM}_2^{(d-1)}}, \hat{y}_i^{\mathrm{SVM}_3^{(d-1)}}) \end{aligned} \tag{28}$$

$$\begin{aligned} \mathbf{x}_i^{\mathrm{SVM}_2^{(d)}} &= (I_1, \hat{y}_i^{\mathrm{SVM}_1^{(d-1)}}, \hat{y}_i^{\mathrm{SVM}_3^{(d-1)}}) \\ &= (\mathbf{x}_i^{\mathrm{SVM}_2^{(0)}}, \hat{y}_i^{\mathrm{SVM}_1^{(d-1)}}, \hat{y}_i^{\mathrm{SVM}_3^{(d-1)}}) \\ &= (x_i^{(1\cdot\frac{l}{3})+1}, \ldots, x_i^{(2\cdot\frac{l}{3})}, \hat{y}_i^{\mathrm{SVM}_1^{(d-1)}}, \hat{y}_i^{\mathrm{SVM}_3^{(d-1)}}) \end{aligned} \tag{29}$$

$$\begin{aligned} \mathbf{x}_i^{\mathrm{SVM}_3^{(d)}} &= (I_2, \hat{y}_i^{\mathrm{SVM}_1^{(d-1)}}, \hat{y}_i^{\mathrm{SVM}_2^{(d-1)}}) \\ &= (\mathbf{x}_i^{\mathrm{SVM}_3^{(0)}}, \hat{y}_i^{\mathrm{SVM}_1^{(d-1)}}, \hat{y}_i^{\mathrm{SVM}_2^{(d-1)}}) \\ &= (x_i^{(2\cdot\frac{l}{3})+1}, \ldots, x_i^l, \hat{y}_i^{\mathrm{SVM}_1^{(d-1)}}, \hat{y}_i^{\mathrm{SVM}_2^{(d-1)}}) \end{aligned} \tag{30}$$

These further inputs $\hat{y}_i^{\mathrm{SVM}_1^{(d-1)}}$, $\hat{y}_i^{\mathrm{SVM}_2^{(d-1)}}$ and $\hat{y}_i^{\mathrm{SVM}_3^{(d-1)}}$ are the outputs, i.e the predicted class labels, of the SVMs $\mathrm{SVM}_1^{(d-1)}$, $\mathrm{SVM}_2^{(d-1)}$, $\mathrm{SVM}_3^{(d-1)}$ of the previous plait layer $P_{(d-1)}$. That means that $\mathrm{SVM}_1^{(d)}$, $\mathrm{SVM}_2^{(d)}$ and $\mathrm{SVM}_3^{(d)}$ take into account the classification decisions – wrong or correct – of the previous SVMs $\mathrm{SVM}_1^{(d-1)}$, $\mathrm{SVM}_2^{(d-1)}$, $\mathrm{SVM}_3^{(d-1)}$ to improve their own classification. The described approach will be proven by experiments in the following section V.

V. Experiments

To prove the proposed approach, we conducted 4 main experiments which will be discussed in section V-C, V-D and V-E. The real data used and the experimental settings are described in section V-A and V-B.

Table I

Numbers (#) of students, of examples (tasks overall) and of examples with class label *over-challenged* as well as of examples with class label *flow* for the German and the English data.

Data set	# students	# examples	# *over-challenged*	# *flow*
German	10	34	12	22
English	6	20	6	14

Table II

Experiments with features from amplitudes. Numbers (#) of train and test examples for the 2-fold cross validation for every subset. In brackets the distribution of both classes is noted: (# *over-challenged* + # *flow*). The last column reports the classification test error of a single SVM.

Data	# train fold 1	# test fold 1	# train fold 2	# test fold 2	Error
German (subset 1)	11 (6+5)	12 (6+6)	12 (6+6)	11 (6+5)	30.30
German (subset 2)	11 (6+5)	12 (6+6)	12 (6+6)	11 (6+5)	30.30
German-English (subset 1)	18 (9+9)	18 (9+9)	18 (9+9)	18 (9+9)	36.11
German-English (subset 2)	18 (9+9)	18 (9+9)	18 (9+9)	18 (9+9)	33.33

A. Data Sets

For the experiments we used two different real data sets (see table I) collected in the course of the EU project iTalk2Learn ([8]). The first data set was gained from interactions with German students and the second one from interactions with English students.

For the German data set a study was conducted in which the speech and actions of ten 10 to 12 years old German students were recorded and students perceived task-difficulties (see section III) were reported. During the study a paper sheet with fraction tasks was shown to the students and they were asked to paint – by means of a software for painting with a computer – and explain their observations and answers. The acoustic speech recordings, consisting of 10 wav files with a length from 15 up to 20 minutes, were used to gain the input features for affect recognition, i.e. the amplitude and articulation features.

For the English data set the speech data of six British students in the age of 8 to 11 years were recorded and the perceived task-difficulties reported. During the study the students were asked to solve fraction tasks of a tutoring system on a computer and to explain their observations and solutions. The acoustic speech recordings for extracting the features for the affect recognition (amplitude features) consist of 6 wav files with a length from 11 up to 30 minutes. As the English data set is quite small and we want to find

Table III
CLASSIFICATION TEST ERRORS OF THE SINGLE SVMS, ENSEMBLE METHODS AND SVM PLAIT APPLIED TO FEATURES FROM AMPLITUDES OF THE 2 GERMAN-ENGLISH SUBSETS AND THE AVERAGE (LAST COLUMN).

	German-English (subset 1)	German-English (subset 2)	avrg.
Single SVM	36.11	33.33	34.72
SVM$_1^{(0)}$	33.33	30.56	31.95
SVM$_2^{(0)}$	38.89	27.78	33.34
SVM$_3^{(0)}$	36.11	33.33	34.72
Majority	36.11	25.00	30.56
Stacking	25.00	25.00	25.00
SVM Plait	**22.22**	**19.44**	**20.83**

out if the features are language independent so that we can generalize our system, we did no experiments only with the English data but with the English data together with the German data.

Overall we conducted 4 main experiments:

(I) a single SVM applied to features from amplitudes of the German data,
(II) a single SVM applied to features from amplitudes of the German-English data,
(III) SVM plait with 3 SVMs in each layer applied to features from amplitudes of the German-English data and
(IV) SVM plait with 3 SVMs in each layer applied to articulation features of the German data.

The results of the 4 main experiments are reported in the following subsections.

B. Experimental Settings

For the experiments we used the library LIBSVM ([3]) delivering an implementation of support vector machines. We applied SVMs with an RBF-kernel and for each SVM used we conducted a grid search (according to [7]) to estimate the optimal values for the hyper parameters C and γ. The input feature values which are fed into the SVMs are normalized to be in the interval $[0, 1]$. As mentioned in section III the SVMs are used to solve a binary classification problem, i.e. to classify examples as either *over-challenged* or *flow*. Because of the small size of the data and as the data is imbalanced (in table I one can see that there are more examples with label *flow*) we applied a variation of a 2-fold cross validation: for every main experiment we split the set of examples with label *flow* into 2 subsets and conducted 2 different experiments consisting of a 2-fold cross validation with each of the 2 subsets. This approach leads to balanced data as one can see in table II.

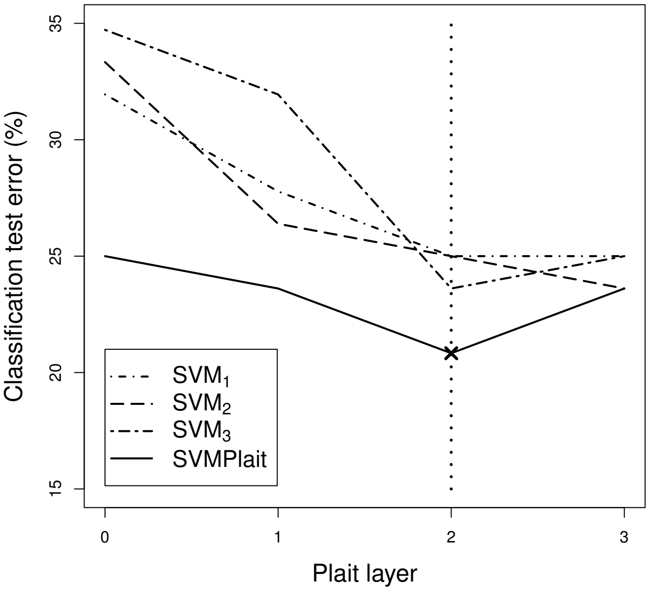

Figure 3. The evolution of the average classification test error of the single SVMs SVM$_1$, SVM$_2$, SVM$_3$ and the SVM plait over the plait layers for the amplitude features of the German-English data.

C. Results of Experiment I and II

For experiment I and II we applied a single SVM to the amplitude features of the German data and the mixed German-English data. The results are shown in table II. The classification test error is smaller for the German data alone than for the mixed German-English data, but the error for the mixed German-English data is still good enough to assume that in cases where the data is too small for training, like our English data, we can enable the training by using further data from a different source, like our German data, and still get a good classification result. However, the classification results with the single SVM overall are not satisfactory enough and the question arises if one could improve the classification performance. Hence, in the next section we investigate the application of the SVM plait to the data.

D. Results Experiment of III

In experiment III we applied the SVM plait with 3 plait layers to the amplitude features of the German-English data. The results are shown in table III. Table III shows the classification test errors for both subsets as well as the average error. The classification test error is reported

(a) for a single SVM (see also experiment II),
(b) for the three single SVMs SVM$_1^{(0)}$, SVM$_2^{(0)}$, SVM$_3^{(0)}$ of the first plait layer, were each of them is applied to one of three overlapping splits of the amplitude feature vector $((x_i^0, \ldots, x_i^3)$, $(x_i^2, \ldots, x_i^5)$, $(x_i^4, \ldots, x_i^7))$,
(c) for two ensemble methods applied to the outputs of SVM$_1^{(0)}$, SVM$_2^{(0)}$, SVM$_3^{(0)}$: majority vote and stacking, i.e. a subsequent SVM,

(d) the SVM plait.

In table III one can see that the SVM plait outperforms the single SVMs as well as the ensemble methods and improves on average the classification performance by 40 % in comparison to the single SVM applied in experiment II.

In figure 3 the evolution of the average classification test error of $\text{SVM}_1^{(d)}$, $\text{SVM}_2^{(d)}$, $\text{SVM}_3^{(d)}$, $d = 0, \ldots, (q+1), q = 2$, and the SVM plait (with $1, \ldots, (q+2)$ layers) over the plait layers is shown. As one can see the error decreases for all of them over the plait layers. In the third plait layer the classification test error of the SVM plait is minimal, hence we applied an SVM plait with 3 plait layers in this experiment.

E. Results of Experiment IV

In experiment IV we applied the SVM plait with 5 plait layers to the articulation features of the German data. The results are shown in table IV. Table IV shows the classification test errors for both subsets as well as the average error. Similar to experiment III the classification test error is reported

(a) for a single SVM applied to the full feature vector $(x_i^0, \ldots, x_i^{12})$,

(b) for the three single SVMs $\text{SVM}_1^{(0)}$, $\text{SVM}_2^{(0)}$, $\text{SVM}_3^{(0)}$ of the first plait layer, were each of them is applied to one of three splits of the amplitude feature vector $((x_i^0, \ldots, x_i^4)$, $(x_i^5, \ldots, x_i^8)$, $(x_i^9, \ldots, x_i^{12}))$,

(c) for two ensemble methods applied to the outputs of $\text{SVM}_1^{(0)}$, $\text{SVM}_2^{(0)}$, $\text{SVM}_3^{(0)}$: majority vote and stacking, i.e. a subsequent SVM,

(d) the SVM plait.

In table IV one can see that the SVM plait outperforms the single SVMs as well as the ensemble methods and improves on average the classification performance by 57 % in comparison to the single SVM applied to the full feature vector.

A comparison of the results of the two single SVMs applied to the full input feature vectors in experiment III and IV shows that on average the classification performance on articulation features is similar good as the classification performance on amplitude features.

In figure 4 the evolution of the average classification test error of $\text{SVM}_1^{(d)}$, $\text{SVM}_2^{(d)}$, $\text{SVM}_3^{(d)}$, $d = 0, \ldots, (q+2), q = 4$, and the SVM plait (with $1, \ldots, (q+3)$ layers) over the plait layers is shown. As one can see the error decreases over the plait layers. In the fifth plait layer the classification test error of the SVM plait reaches its minimum, hence we have applied an SVM plait with 5 plait layers in this experiment.

VI. Conclusions and Future Work

We presented an approach for improving the affect recognition in intelligent tutoring systems by using a plait structure of support vector machines with different input feature vectors. Furthermore, we proposed a new kind of features for that problem stemming from initial processing steps of speech recognition. The results of our experiments with real data sets show that the proposed SVM plait approach is able to improve the classification performance significantly and that the proposed amplitude and articulation features are suitable for affect recognition in intelligent tutoring systems. Some future work will be to investigate how to combine both feature types.

Table IV
Classification test errors of the single SVMs, ensemble methods and SVM plait applied to articulation features of the 2 German subsets and the average (last column).

	German (subset 1)	German (subset 2)	avrg.
Single SVM	39.02	31.06	35.04
$\text{SVM}_1^{(0)}$	26.14	30.30	28.22
$\text{SVM}_2^{(0)}$	39.02	39.02	39.02
$\text{SVM}_3^{(0)}$	48.11	39.77	43.94
Majority	34.78	34.78	34.78
Stacking	26.14	39.39	32.77
SVM Plait	**8.33**	**21.59**	**14.96**

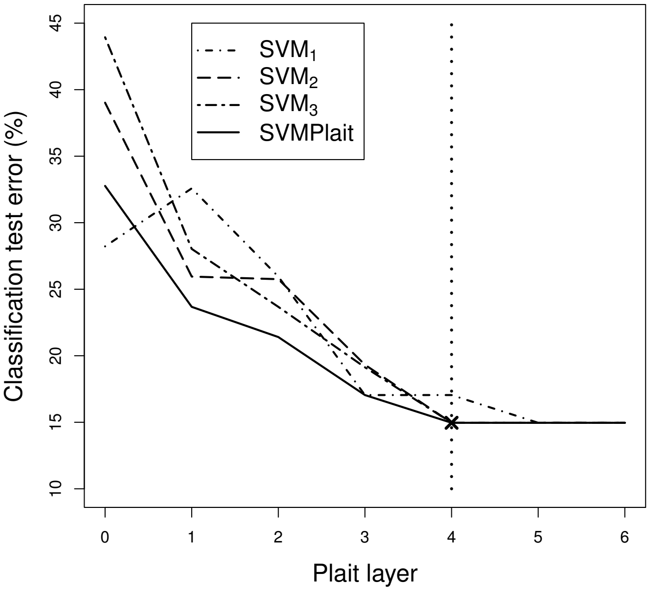

Figure 4. The evolution of the average classification test error of the single SVMs SVM_1, SVM_2, SVM_3 and the SVM plait over the plait layers for the articulation features of the German data.

Moreover, with this work we have shown that the general plait structure does not just work with artificial neural networks like convolutional neural networks or multilayer perceptrons within the HNNP but also with support vector machines. Hence, some future work would be to investigate, if the plait structure also works with other classifiers and to define a general plait principle.

Acknowledgment

The research leading to the results reported here has received funding from the European Union Seventh Framework Programme (FP7/2007 – 2013) under grant agreement no. 318051 – iTalk2Learn project ([8], www.italk2learn.eu). Furthermore, we thank our project partners Ruhr University Bochum and Whizz Education for realising the data collection.

References

[1] O. Abdel-Hamid, A. Mohamed, H. Jiang and G. Penn, *Applying Convolutional Neural Networks concepts to hybrid NN-HMM model for speech recognition*, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4277–4280, 2012.

[2] B. E. Boser, I. Guyon and V. Vapnik, *A training algorithm for optimal margin classifiers*, Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144–152, ACM Press, 1992.

[3] C. C. Chang and C. J. Lin, *LIBSVM: A library for support vector machines*, ACM Transactions on Intelligent Systems and Technology, Vol. 2(3), pp. 1–27, 2011. (software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm)

[4] A. Cichocki, R. Zdunek, A. H. Phan and S. I. Amari, *Non-negative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*, Wiley, 2009.

[5] C. Cortes and V. Vapnik, *Support-vector network*, Machine Learning, 20:273–297, 1995.

[6] H. P. Graf, E. Cosatto, L. Bottou, I. Durdanovic and V. Vapnik, *Parallel support vector machines: The cascade svm*, Advances in Neural Information Processing Systems, pp. 521–528, MIT Press, 2005.

[7] C. W. Hsu, C. C. Chang and C. J. Lin, *A Practical Guide to Support Vector Classification*, Technical report, Department of Computer Science, National Taiwan University, 2011. (http://www.csie.ntu.edu.tw/ cjlin/)

[8] *Talk, Tutor, Explore, Learn: Intelligent Tutoring and Exploration for Robust Learning (iTalk2Learn)*, http://www.italk2learn.eu.

[9] R. Janning, A. Busche, T. Horváth and L. Schmidt-Thieme, *Buried Pipe Localization Using an Iterative Geometric Clustering on GPR Data*, Artificial Intelligence Review, DOI: 10.1007/s10462-013-9410-2, Springer, 2013.

[10] R. Janning, C. Schatten and L. Schmidt-Thieme, *HNNP – A Hybrid Neural Network Plait for Improving Image Classification with Additional Side Information*, Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI) 2013, pp. 24–29, Washington DC, USA, 2013.

[11] R. Janning, C. Schatten and L. Schmidt-Thieme, *Automatic Subclasses Estimation for a Better Classification with HNNP*, Proceedings of the 21th International Symposium on Methodologies for Intelligent Systems (ISMIS 2014), in Lecture Notes in Artificial Intelligence, Springer, 2014.

[12] R. Janning, C. Schatten and L. Schmidt-Thieme, *Multimodal Affect Recognition for Adaptive Intelligent Tutoring Systems*, Workshop on Feedback from Multimodal Interactions in Learning Management Systems (EDM 2014), 2014.

[13] R. Janning, C. Schatten and L. Schmidt-Thieme, *Feature Analysis for Affect Recognition Supporting Task Sequencing in Adaptive Intelligent Tutoring Systems*, European Conference on Technology Enhanced Learning (EC-TEL 2014), 2014.

[14] R. Janning, C. Schatten and L. Schmidt-Thieme, *Local Feature Extractors Accelerating HNNP for Phoneme Recognition*, Proceedings of the 37th German Conference on Artificial Intelligence (KI 2014), in Lecture Notes in Artificial Intelligence, Springer, 2014.

[15] S. Luz, *Automatic Identification of Experts and Performance Prediction in the Multimodal Math Data Corpus through Analysis of Speech Interaction*, Second International Workshop on Multimodal Learning Analytics, Sydney, Australia, 2013.

[16] S. K. D'Mello, S. D. Craig, A. Witherspoon, B. McDaniel and A. Graesser, *Automatic detection of learner's affect from conversational cues.*, User Model User-Adap Inter, DOI 10.1007/s11257-007-9037-6, 2008.

[17] J. D. Moore, L. Tian and C. Lai, *Word-Level Emotion Recognition Using High-Level Features*, Computational Linguistics and Intelligent Text Processing (CICLing 2014), pp. 17–31, 2014.

[18] L. P. Morency, S. Oviatt, S. Scherer, N. Weibel and M. Worsley, *ICMI 2013 grand challenge workshop on multimodal learning analytics*, Proceedings of the 15th ACM on International conference on multimodal interaction (ICMI 2013), pp. 373–378, 2013.

[19] C. Schatten and L. Schmidt-Thieme, *Adaptive Content Sequencing without Domain Information*, Proceedings of the Conference on computer supported education (CSEDU 2014), 2014.

[20] B. Schuller, A. Batliner, S. Steidl and D. Seppi, *Recognising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge*, Speech Communication, Elsevier, 2011.

[21] K. M. Ting and I. H. Witten, *Issues in stacked generalization*, Journal of artificial intelligence research, Vol. 10, pp. 271–289, 1999.

[22] L. L. S. Vygotsky, *Mind in society: The development of higher psychological processes*, Harvard university press, 1978.

[23] M. Worsley and P. Blikstein, *What's an Expert? Using Learning Analytics to Identify Emergent Markers of Expertise through Automated Speech, Sentiment and Sketch Analysis*, Proceedings of the 4th International Conference on Educational Data Mining (EDM '11), pp. 235–240, 2011.