Analysis of Dysfluencies by Computational Intelligence

Juraj Pálfy*
Institute of Applied Informatics
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava 4, Slovakia
palfy@fiit.stuba.sk

Abstract
The work presents a contemporary perspective of the field of dysfluent speech processing. It deals with suitable approaches of computational intelligence for speech analysis. It presents computational intelligence classifiers, based on probabilistic decision theory and classifiers based on cost function optimization. This work deals with common model of automatic speech recognition and explains the historical and the current speech feature extraction methods. It introduces definitions and known symptoms of communication disorder, stammering, and selects the symptoms that are the subject of our interest and describes the state-of-the-art of computer analysis in stuttered speech. In our experiments we deal with Artificial Neural Networks and with Support Vector Machines, to solve the task of recognizing simple dysfluencies. We were looking for inspiration in data mining and in bioinformatics, to handle the problem of detecting complex dysfluent events. The proposed new methodology for processing long dysfluent speech intervals was based on the speech signal transformed into symbolic sequences. We present new algorithms, able to automatically adapt the length of the analysis window and detect complex dysfluencies. Moreover we found that dysfluencies have recursive property which was used in developing our algorithms. We derived new features of dysfluencies from yields of our new algorithms. We borrow inspiration from the field of video sequence analysis, to apply and derive new functions designated to detect specific dysfluent events. The new methodology for symbolic analysis of dysfluent speech, the new features of complex dysfluencies and the new algorithms were related to established methodologies and statistically compared.

Categories and Subject Descriptors
I.2.1 [Artificial Intelligence]: Applications and Expert Systems—medicine and science; I.2.7 [Artificial Intelligence]: Natural Language Processing—speech recognition and synthesis; I.5.2 [Pattern Recognition]: Design Methodology—classifier design and evaluation, feature evaluation and selection; I.5.4 [Pattern Recognition]: Applications—signal processing; K.4.2 [Computers and Society]: Social Issues—assistive technologies for persons with disabilities

Keywords
computational intelligence, data mining, bioinformatics, speech, stammering.

1. Introduction
Various methods of pattern detection in time series are already used in Data Mining and Bioinformatics. Speech is only a different manifestation of time series, where dysfluencies may disrupt the information flow in talkers communication channel. In case of stuttered speech, the dysfluencies are more frequent than in fluent talkers, hence it is ideal for observation. According to Speech Language Pathology, repetitions and prolongations are assigned to the set of dysfluent symptoms.

Common dysfluencies like prolongations, repetitions and hesitations add needless lexical information to talkers’ conversation. Information redundancy, caused by common dysfluencies, negatively influence the Automatic Speech Recognition (ASR) system performance. Therefore one of many aspects of dysfluency detection in speech technology, is to augment the ASR system to decrease the recognition errors.

Stuttering is characterized by a set of clinical symptoms, namely: excess effort, dysfluencies and psychic tension [20]. Symptoms (Fig. 1) in excess effort are visible, psychic tension contains symptoms which are hidden from an observer (like negative emotions, cognitive activities, etc.). We are interested in dysfluencies, which are in the category of audible symptoms. World Health Organization, identifies stuttering with code F98.5 and defines stuttering as a speech characterized by frequent repetition or prolongation of sounds or syllables or words, or by frequent hesitations or pauses [10]. Stuttered speech because of its features, may be interesting in research where emotional or cognitive processes are studied, because these are more time consuming to observe in a spontaneous fluent speech.

---

*Recommended by thesis supervisor: Prof. Jiří Pospíchal. Defended at Faculty of Informatics and Information Technologies Slovak University of Technology in Bratislava on June 25, 2014.

© Copyright 2014. All rights reserved. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from STU Press, Vazovova 5, 811 07 Bratislava, Slovakia.

This paper provides an introduction to the theoretical background for anatomy and physiology of speech and hearing; the conventional algorithms and methods for speech processing, speech feature extraction and automatic speech recognition are recommended. The technical documentation of frequent dysfluencies (e.g. syllable repetitions, phrase repetitions, prolongations, etc.), novel observation of complex dysfluencies in stuttered speech and our examination of dysfluencies were presented.

In this paper we focus on basic principles needed to briefly introduce, understand and clarify the methodologies together with our own contributions.

In the next two sections (Subsection 1.1 as well as Subsection 1.2), we describe the motivation including the status quo of dysfluency recognition. After reporting the issues about dysfluencies in Section 2, we portray two computational intelligence approaches for dysfluency recognition, the Artificial Neural Networks (ANN) in Section 2.2 and the Support Vector Machines (SVM) in Section 2.3. In Section 4, we report the experiment results based on improved common methodology for recognition of simple dysfluencies. In the last, experimental Section 5 we provide solution for detecting complex dysfluencies. Afterwards we discuss the main outcomes along with conclusion of our findings in Section 7 as well as the consequence of testing the principal hypothesis (see Section 3, p. 3).

1.1 Motivation: Why Recognize Dysfluencies?

Dysfluent speech recognition gets a lot of attention in field of Speech Language Pathology (SLP), where objective evaluation of stuttered speech is still under development. Researchers with technical background invest their effort to develop framework to resolve this problem for SLP. Authors [25] on basis of spectral changes and Bayesian detector define a parameter for global evaluation of dysfluent speech, to define an effectiveness of SLP therapy. Phoneme repetitions were classified by Hidden Markov Models on basis of Mel-Frequency Cepstral Coefficients (MFCC) in [48]. MFCC with Dynamic Time Warping (DTW) were used by [35] to study syllable repetition detection accuracy with various multidimensional MFCC feature vectors (with 12, 13, 26 and 39 dimension). [15] examined the performance of Least Square Support Vector Machines with Sample Entropy derived from Bark scale, Erb scale and Mel scale for distinguishing the prolongations and repetitions in speech. MFCC and Linear Predictive Cepstral Coefficients feature extractions with classifiers k-Nearest Neighbor and Linear Discriminant Analysis were compared to recognize repetitions and prolongations in stuttered speech. [42] introduced hierarchical Artificial Neural Networks to support the stuttered speech recognition process.

Another aspect for dysfluent speech recognition is to moderate its effect in ASR. Researchers effort in this case, is to improve (with the dysfluency detection module) the accuracy in current ASR, by introducing information for ASR about the unknown phenomenon. In [36] the speech dysfluencies were studied in case of fluent (i.e. normal or non-pathological) speech. Early detection of dysfluencies studied in [24], shows that the juncture phenomena which occur between words in fluent speech are usually absent at the interruption point in dysfluent utterances. Textual and prosody information was used in [23] with Conditional Random Fields to locate interruption point, point in time at which the speaker breaks off from the original utterance. Similar feature extraction technique, but on basis of Weighted Finite State Transducers to [23] was implemented to detect filled pauses and reparandum, the region of repeats and repairs [27].

1.2 Status Quo: Classifiers of Computational Intelligence for Dysfluency Recognition

Published works in speech recognition of dysfluencies use classifiers. Classifiers according to Table 1 basically fall into two groups. Firstly, the group, which includes classifiers based on decision trees (also known as statistical models):

- Naive Bayes
- Hidden Markov models (HMM)

Secondly classifiers, working also with optimizing their cost functions:

- Artificial neural networks (ANN)
- Support vector machines (SVM)

Works included in the Table 1 are in chronological order of publication. In the first column are listed first authors of the research teams. Year of publication is in the second column. In the third column are properties such as...
frequency, in which sound recordings were processed, or algorithms for obtaining set of speech feature vectors.

After analysis of the works referenced in the Table 1, we have indicated general stuttered speech analysis processes (Figure 1). In the first step (Figure 1) stuttered speech record enters the segmentation phase. In the segmentation stage, the audio recording is divided into smaller parts. In the third step, the dimensionality of speech features vectors (also called parameters of speech) is reduced. Thereafter, the speech feature vectors enter the classification. The quality of the classifier depends on the vector properties of speech, which are used in decisions.

2. Technical Consideration and Problems with Dysfluencies

Dysfluencies are divided into subcategories, where each dysfluent event has its annotation label (e.g. RI denotes syllable repetition re re research or P denotes prolongation rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr
and shortened the run time by 10%. Instead of direct transformation to symbolic sequences, firstly a short time energy was computed and then transformed to symbolic representation of speech (with alphabet size 10 and word size 500). The amount of analyzed speech data was increased 10 times. Apart from the further developed algorithm, we present a wider statistical analysis of new dysfluency detection functions and a comparison of standard feature extraction method with method based on symbolic sequences.

### 2.1 Issues in Annotation

During our first annotation session, the fundamental question arose: how to denote dysfluent intervals in the process of annotation? Speech Language Pathologists (SLPs) when doing stuttered speech analysis, commonly do not need the lexical content of audio record to be aligned to time axis. In case of repeated words see Fig. 3, the whole interval from the first occurrence of \( \text{if} \) until the last occurrence \( \text{if} \) would be annotated by label \( \text{RS} \) (in slovak repetica slova). In domain of dysfluent speech recognition they do not deal with this issue. Through the technical processing, the entire interval where dysfluencies occur, should be further divided for the purpose of supplementary processing (e.g. marked dysfluency in case of prolongations to be shorten, repetitions to be deleted, etc.). During the early annotations, we intuitively considered in case of repetitions (e.g. word repetition) the first occurrence \( \text{if} \) as correctly spoken word and its subsequent repetition as the dysfluent part (i.e. repetition, which eventually should be removed in post-processing). In the literature survey, where papers have dealt among other topics, with cognitive processes in the speech development and observing other types of dysfluencies, we came to the conclusion that dysfluencies should be divided and annotated just the vice versa.

The first occurrence of \( \text{if} \) is a repeated word and the last occurrence of \( \text{if} \) is a properly realized/spelled word. Speech Language Pathologists in the case described in Fig. 3 annotate the entire interval in the annotated text \( \text{fo fo forty-four} \) as syllable repetition, \( \text{R1} \) according to our convention, in this dysfluent interval repeated syllables \( \text{fo fo} \) are prefixes and then the remaining suffix of word \( \text{forty-four} \) is a correctly realized word. This new perspective applies to a great extent for complex dysfluencies. In our algorithms the application of suffix arrays in symbolic representation of speech searches for \textit{recursion of symbolic sequences}. We proposed method on basis of oriented graphs for graphical modeling of specific types of dysfluencies (Fig. 5). On oriented graph, states (i.e. vertexes) are denoted by \( s_0 \ldots s_n \). Oriented edges (i.e. paths between vertexes) of oriented graph, are labeled by \( e_0 \ldots e_n \). Then for concrete example (Fig. 3) the word repetition begins in state \( s_0 \) (Fig. 5 b)), in state \( s_1 \) occurs silent pause \( \text{RS} \) with lexical content \( \text{if} \), after that we proceed to state \( s_2 \) containing word repetition, afterwards in state \( s_3 \) takes place another silent pause, the second \( \text{if} \) is assumed to be realized correctly, therefore we termiate our graphical model (Fig. 5 b)) at state \( s_n \) (and go ahead to realize if \( \text{he buy} \)).

During the manual annotation, we encountered of such examples, where several types of dysfluent events were not listed in the commonly/classically used SLP classification from Shipley, McAfee (1998, in [20]). We eventually collected from various sources and identified these primarily unspecified dysfluent events:

- \( \text{RZ} \) - sound repetition (Gregory, 1992 in [2])
- \( V \) - hesitation, e.g. triplet - reparandum, edition and correction
- \( \text{KD} \) - complex dysfluencies [36]

During annotation we often have to address the problem: how to annotate type of dysfluency, which is in the half way between the prolongation and between repetition of syllables, but not in common SLP classification? \( \text{RZ} \), at the first hearing is similar to a prolongation, with repeated equally short sounds, which are shorter as phonemes, however it is audible that between \( \text{RZ} \) occur short silent pauses, which are characteristic of the repetitions.

In speech processing, we gathered a special type of hesitation \( V \), namely a triplet \( < \text{reparandum} > < \text{edition} > < \text{correction} > \), \( V \) was the most complex specimen of dysfluency technically documented at the signal level and accessible in literature. Shipley and McAfee in (1998) has not considered \( V \), it is the instance of Slovak SLP’s convention, which is based on Shipley and McAfee (1998). Augmented categories of dysfluencies were documented in [44], here \( V \) was considered, though its special type/form (a triplet) is not concluded. According to our research, we are the first who identified, used and complemented to (our internal) SLP convention, this special type of hesitation (a triplet) \( V \). In psychology [36], a dissertation has been written about the theory of dysfluencies. Its author, Mrs Elizabeth E. Shriberg (1994, pp. 165 – 174, [36]) in accordance with our observations of the stuttered speech, also observed complex dysfluencies, but in speech of people without communication disorder. She made her study [36] in the SWITCHBOARD corpus, which con-

### Table 2: Summary of the general approach of recognizing dysfluencies, problems with dysfluencies along with outline of our proposals for addressing these challenges.

<table>
<thead>
<tr>
<th>General approach</th>
<th>Problems with dysfluencies</th>
<th>Proposal of solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>short intervals fixed window simple dysfluences feature extraction classification</td>
<td>long intervals dynamic distribution in time complex dysfluencies sparse occurrence statistical modeling</td>
<td>symbolic representation of speech dynamic window adaptation observed recursive property custom algorithms/functions custom features detection of complex dysfluencies</td>
</tr>
</tbody>
</table>
Complex dysfluencies (we use label for that *KD*, Fig. 4) are characterized by being composed of rich chaotic mixtures (i.e., combinations) of simple (i.e., fundamental) dysfluent events (in Fig. 4 take place hesitation, word repetition, and prolongation).

2.2 Artificial Neural Networks

In this section we introduce two of the Neural Network family of classifiers that are discussed later in the section of results. A neural network is characterized by its prominence on using many interconnected processors that perform relatively slow and individual calculations in parallel [18]. It allows using very simple computational operations (additions, multiplications and fundamental logic elements) to solve complex, mathematically ill-defined problems, nonlinear problems or stochastic problems [11].

Multi-layer Perceptron (MLP) is a feedforward neural network with one or more hidden layers, whose computation nodes are called hidden neurons. The function of hidden neurons is to intervene between the external input and the network output in some useful manner. By adding hidden layers, the network is enabled to extract higher-order statistics. This tendency of hidden neurons is particularly valuable when the dimension of the input layer is large [17], p. 43.

In Elman network a set of context units are introduced, which are extra input units whose activation values are connected back from the hidden units. The hidden units of the output units are connected back and the extra input units have no self-connections. Again the hidden units are connected to the context units with a fixed weight of value +1 [19], p. 48. The context units give the Elman networks a kind of decaying memory, which has proven sufficient for learning temporal structure over short distances, but not generally over long distances (Servan-Schreiber et al. 1991). These networks can be trained with classical backpropagation, considering all of the trainable weights are feedforward weights [45].

2.3 Support Vector Machines

Since 1995, Support Vector Machines have become one of the preeminent machine learning paradigms. Support Vector Machines are now employed routinely in areas that range from handwriting recognition to bioinformatics to the mining of very large databases [12].

Support Vector Machine can provide a great generalization performance on pattern recognition problems but it does not cumulate problem-domain knowledge [17], p. 340.

Support vector machines and their variants and extensions, often called kernel-based methods (or simply kernel methods), have been studied extensively and applied to various pattern classification and function approximation problems [1].

An SVM is an abstract learning machine, which will learn from training data and attempt to generalize and make correct predictions on testing data [5]. The main part of SVM classifier design is the notion of the margin [46]. The argument inside the decision function of a classifier is

\[ w \cdot x + b = 0 \]  

(1)
functions, the inner product between the representation of two data vectors is defined as 

\[ \langle x, y \rangle = x^T y + r \]  

where \( x \) and \( y \) are vectors of input data, \( r \) is a bias term, \( x^T \) is the transpose of \( x \), and the inner product \( \langle \cdot, \cdot \rangle \) is the scalar product. The separating hyperplane corresponding to Equation 1 is shown as a continuous line in Figure 6. This hyperplane separates the two classes of data with points on the left side labeled \( y_i = -1 \) (Equation 2) and points on the right side labeled \( y_i = +1 \) (Equation 3).

\[
w \cdot x + b \geq 0 \quad \text{(Equation 2)}
\]

\[
w \cdot x + b < 0 \quad \text{(Equation 3)}
\]

and points outside the margin on the correct side of the classifier. \( \xi > 0 \) for points outside, or for points outside the margin on the wrong side of the classifier [46]. The constant \( C \), called the cost, allows us to control the compromise between margin size and error [12].

SVM techniques for solving nonlinear classification tasks use mapping support vectors in areas with higher dimension. In higher dimensions, used by SVM, the classes are likely to be linearly separable. Mapping is as follows:

\[
x \mapsto \phi(x) \in H,
\]

where \( H \) dimension is greater than \( R^n \). If we carefully choose one mapping function from a known family of functions, the inner product between the representation \((\phi(x_1), \phi(x_2))\) points can satisfy the condition

\[
\langle \phi(x_1), \phi(x_2) \rangle = k(x_1, x_2).
\]

Angled brackets \( \langle \cdot, \cdot \rangle \) indicate scalar multiplication in \( H \) and \( k(\cdot, \cdot) \) is a kernel function. Radial basis functions (RBF) of kernel function is defined as

\[
k(x, y) = \exp \left( -\frac{||x - y||^2}{\sigma^2} \right)
\]

where \( \sigma \) is a parameter determined by [46]. The purpose of the user-defined parameter \( \sigma > 0 \) which specifies width, is explained in [40], p. 10. Another kernel function, which we used and compared in our experiment, was sigmoid kernel function,

\[
k(x, y) = \tanh \left( \gamma x^T y + r \right)
\]

with suitably chosen values of \( r \in R \) it led to SVM classifiers with very similar accuracies and SVM sets [37].

\( K \)-fold cross-validation was described by Hamel [12]. A suitable compromise between the potential diversion hold-out method and computational complexity of leave-one-out method is \( K \)-fold cross validation. Suppose \( D \) is our training set (4th column in Table 3).

\[
D = \{(x_1, y_1), \ldots, (x_l, y_l)\} \subset R^n \times \{+1, -1\}
\]

Divide \( D \) in to \( K \) (in our case \( K = 4 \)) partition with \( K \ll l \) as follows

\[
D = Q_1 \cup Q_2 \cup \cdots \cup Q_{K-1} \cup Q_K
\]

For \( Q_i \cap Q_j = \emptyset \), it is true that

\[
|Q_i| = |Q_j| = l/K
\]

where \( i, j = 1, \ldots, K \) and \( i \neq j \). We used every partition for testing only once. The rest of partitions we used to train the model. Let \( Q_i \) be a partition set \( D \), then we construct the corresponding training set

\[
P_i = D - Q_i, \quad \text{for } i = 1, \ldots, K.
\]

\( K \)-fold cross validation can also be used advantageously for large datasets.

3. Goals

Our efforts were focused on these specific problems in dysfluent speech recognition.

- In selected methods of computational intelligence, find possible enhancement for analysis of dysfluences
- Solve the problem of speech representation beneficial to facilitate speech data analysis in several seconds long intervals of speech (i.e. long-term analysis of the speech)
- Search for complex dysfluencies (combination of several symptoms of dysfluency) in long intervals of speech
- Demonstrate applicability of speech transformation to symbolic sequences for searching repeated patterns in speech
- Statistically compare of Mel-Frequency Cepstral Coefficients (MFCC) features, derived MFCC features and our proposed features of dysfluencies
- Design gradual classifier for objective evaluation features of dysfluencies
- Test the hypothesis: Search in long intervals of symbolic sequences produces similar results as the search in long repeated intervals in the spectral domain of speech.
- Compare memory and computational time (i.e. time complexity analysis) necessary to Dynamic Time Warping algorithm and our algorithms
- Develop technical documentation of frequent dysfluencies
4. Methodology for Simple Dysfluencies

Experiments were carried out on unified hardware that was equipped with Intel®Core™i7 Dual T9400@2.53 GHz processor and has 4 GB of memory. Under Linux operating system the Matlab environment was used, in which we conducted our experiments and developed our algorithms. In addition to basic Matlab environment, we used for experiments the freely available libSVM [6] library containing implementation of various SVM kernel functions. For advanced annotation (i.e. text) processing of stuttered speech, the SQLite database was used. Freely available Matlab Toolbox, voicebox帮助 us in speech analysis. MFCC vectors were calculated in Matlab using mfcc() function that was implemented by [9] which provides the possibility of calculating the MFCC feature vectors analogous to the MFCC vectors calculated using the commonly used Hidden Markov Model Toolkit (HTK) in automatic speech recognition. The compatibility of implemented mfcc() function [9] ensure, that our results based on MFCC features are comparable with those obtained by HTK speech recognition framework.

4.1 Database

Comparability of results has an important role in the world of scientific research. The online freely accessible stuttering speech database uClass created by Mr. Howell et al. [13] allows to compare the results of works around the world, dealing with stuttered speech. Most records in UCLASS contains speech of school-age children. Descriptions of recordings and other additional information of speakers, who participated in recordings, are in UCLASS version 1 and 2. UCLASS version 1 only contains monologues. UCLASS version 2 along with monologues has also, read text and spontaneous conversation. The database is available in many optional data formats (e.g. wav, mp3, SFS, CHILDES, Praat). For some audio files is also included perceptive evaluation the quality of record.

A major percentage of publications, from top conferences and journals use UCLASS database. Therefore, we utilized it as well.

4.2 Evaluation

We can also characterize the performance of a model in terms of its accuracy. Mean Square Error (MSE) expresses the training or testing error.

\[
\text{Accuracy} = \left( \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \right) \cdot 100\% \quad (13)
\]

The TP called true positives and TN we know as true negatives, these symbols indicate the correct classifications. FP false positive or FN false negative, express instance

Table 3: Selected data of patients from the UCLASS (Source: [31, 30]).

<table>
<thead>
<tr>
<th>Gender</th>
<th>Age (number)</th>
<th>Dysfluent/fluent segments Number Training Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male (8)</td>
<td>8 - 47</td>
<td>20/8</td>
</tr>
<tr>
<td>Female (8)</td>
<td>9 - 17</td>
<td>20/8</td>
</tr>
</tbody>
</table>

when the classifier’s decision is incorrect.

\[
\text{MSE} = \frac{1}{\tilde{l}} \sum_{i=1}^{\tilde{l}} (f(x_i) - y_i)^2 \quad (14)
\]

Here \( \tilde{l} \) is the number of observation. Observed values \( y_i \) have associated predicted values \( f(x_i) \). Last column with Squared Correlation Coefficient (SCC) contains the square of the correlation coefficients data. SCC is calculated with the help of these coefficients

\[
b = \frac{\tilde{l} \sum_{i=1}^{\tilde{l}} x_i y_i - \sum_{i=1}^{\tilde{l}} x_i \sum_{i=1}^{\tilde{l}} y_i^2}{\tilde{l} \sum_{i=1}^{\tilde{l}} x_i^2 - (\sum_{i=1}^{\tilde{l}} x_i)^2}
\]

\[
b' = \frac{\tilde{l} \sum_{i=1}^{\tilde{l}} x_i y_i - \sum_{i=1}^{\tilde{l}} x_i \sum_{i=1}^{\tilde{l}} y_i^2}{\tilde{l} \sum_{i=1}^{\tilde{l}} y_i^2 - (\sum_{i=1}^{\tilde{l}} y_i)^2}
\]

The correlation coefficient \( r \) is then computed by

\[
r^2 = bb'
\]

The correlation coefficient is also known as the product-moment coefficient of correlation.

For the performance of Neural Network models and Support Vector Machine models, we compared the results according to their average accuracy. Using Mean Squared Error we calculated average errors of neural networks and support vector machines models.

4.3 Recognition of Dysfluencies by Computational Intelligence

In the works dealing with dysfluency detection, the authors rarely define which from the general category of a broad range of dysfluent events were considered in their dysfluency recognition methodology. Prolongations at the initial phoneme of words (e.g. mmmama), or repetitions at the initial part of a word (e.g. de de development) in initial phoneme of words (e.g. mmmama), or repetitions later.

We firstly deal with simple dysfluencies, like prolongations and repetitions, and address the complex dysfluencies later.

Table 3 clearly characterizes the gender, the age range and the number of selected dysfluent and fluent events from the UCLASS database. In accordance with the authors [43], we have prepared the training and test data (in total 28 + 28 = 56 segments).

Similarly to the authors of the Table 1 we started recognizing dysfluent speech with the application of classifiers known in computational intelligence. From the available
In the following sections, we describe a solution for the detection of dysfluencies dynamically changing its length in the symbolic sequences of speech. We discuss and compare our methodology with established, frequently used methodologies in the field of dysfluent speech recognition.

5. Methodology for Complex Dysfluencies

In work [13] the authors used 12 selected audio recordings “working set” from University College London Archive of Stuttered Speech (UCLASS). We used subset of this working set with 22 050 Hz sampling rate and a total of 19:32 min playing time for our experiments.6

5.1 Speech Representation and Feature Extraction

Many works dealing with dysfluency detection use Fourier transformation to analyze spectrum and compute derived spectral features, for example MFCC, LPCC, PLP [33]. Symbolic Aggregate Approximation (SAX) discretization allows a time series of arbitrary length \( n \) to be reduced to a string of arbitrary length \( w \), \( w < n \), typically \( w << n \). The alphabet size is an integer \( a \), which satisfies \( a > 2 \). As an intermediate step between the original time series and its SAX representation, we must create a dimensionality reduced version of the data [22]. We utilize the Piecewise Aggregate Approximation (PAA) [21], [47]. Let time series \( X \) of length \( n \) is represented by a reduced vector

\[
X = x_1, \ldots, x_N. \tag{18}
\]

The \( i \)th element of \( X \) is calculated by:

\[
\pi_i = \frac{N}{n} \sum_{j=i}^{n} x_j. \tag{19}
\]

In order to reduce the data from \( n \) dimensions to \( N \) dimensions, the data is divided into \( N \) equal windows. The mean value of the data falling within a window is calculated and a vector of these values becomes the data reduced representation [21]. SAX produces symbols with equiprobability, since normalized time series have a Gaussian distribution. With Gaussian distribution, we can determine the breakpoints that will produce equivalent areas under Gaussian curve. Breakpoints are a sorted list of numbers

\[
B = \beta_1, \ldots, \beta_{n-1} \tag{20}
\]

such that the area under an \( N(0,1) \) Gaussian curve from \( \beta_i \) to \( \beta_{i+1} \) equals \( \frac{1}{B} \) [22]. Symbol concatenation, that represents a subsequence, announces a word. A subsequence

---

6Further information about data and developed algorithms: http://sites.google.com/site/georgepalfy/.
Figure 7: Converting dysfluent speech (contains: “c can c c can”) to symbolic sequences (Source: [29]).

Figure 8: Suffix array construction for string $C = \text{“processing”}$: $\text{Pos}[i]$ denotes a constructed suffix array, $C[\text{Pos}[i]...n]$ lists all prefixes in an array of suffixes (Source: [29]).

$S$ of length $n$ can be represented as a word

$$W = \tilde{w}_1 \ldots \tilde{w}_m.$$  \hspace{1cm} (21)

Let $a_i$ be the $i$th symbol of the alphabet, i.e. $a_3 = c$, $a_4 = d$. The mapping from a PAA approximation $X$ to a word $W$ is obtained by:

$$\tilde{w}_i = a_i, \text{ iff } \beta_{j-1} < w_j \leq \beta_j.$$ \hspace{1cm} (22)

SAX enables to optimally reduce the size of the speech signal and represent a signal as a set of ordered symbolic sequences.

5.2 Data Structure

Data structure of our algorithm is inspired by bioinformatics, where it is used in DNA analysis. A primary motivation for suffix arrays was to enable efficient execution of on-line string queries for very long genetic sequences (for example an order of one million or greater symbols long) [26]. Consider a large sequence

$$C = c_0 c_1 \ldots c_{N-1}$$ \hspace{1cm} (23)

of length $N$. The suffix of $C$ that begins at position $i$, shall be expressed by

$$C_i = c_i c_{i+1} \ldots c_{N-1}.$$ \hspace{1cm} (24)

The basis of this data structure is a lexicographically sorted array. $\text{Pos}$, of the suffixes of $C$. $\text{Pos}[k]$ is the onset position of the $k$th smallest suffix in the set

$$C_0, C_1, \ldots C_{N-1}.$$ \hspace{1cm} (25)

We assume that $\text{Pos}$ is given.

$$C_{\text{Pos}[0]} < C_{\text{Pos}[1]} < \cdots < C_{\text{Pos}[N-1]},$$ \hspace{1cm} (26)

where $<\cdot$ denotes the lexicographical order [26]. Figure 8 provides an example for suffix array construction.

5.3 Functions for Detecting Minimal Signal Change

We assume that prolongations are characterized by minimal difference between $n$ neighboring frames of prolonged speech. We hope to capture these frames using special functions from 2D equations for video segmentation [4], detecting repeated signal interval. We derived functions $D(x, y)$, $D_b(x, l)$, $D_l(H_x, l)$, which were specially adapted for speech [28]. Let vectors

$$x = x_1, \ldots x_N, \quad y = y_1, \ldots y_N,$$ \hspace{1cm} (27)

be the vectors of length $N$, then function

$$D(x, y) = \frac{1}{N} \sum_{i=1}^{N} |x_i - y_i|,$$ \hspace{1cm} (28)

computes vectors difference. The function

$$D_b(x, l) = \sum_{i=1}^{l} D(x_i, x_{i+l}),$$ \hspace{1cm} (29)

measures the difference between vectors $x$ and $x + l$ in the speech, where $b$ is the number of blocks, the parameter $l$ denotes an offset. The value of $l$ is a tradeoff between the detection of abrupt changes ($l = 1$) or smooth transitions ($l > 1$). Let $b$ be the number of $H$ histograms with identical classes. The last function

$$D_l(H_x, l) = \sum_{i=1}^{l} D(H_x(i), H_x(i + l)),$$ \hspace{1cm} (30)

with two input parameters: $H_x$ histograms of vectors $x$ and offset $l$ computes the difference between distribution of vectors without taking into account their position. Articulation organs in vowel realization emit periodic signal. Our derived functions give us an acceptable score (Figure 9), characterizing a minimal change in these periodic signal changes along analysis windows. Problem in this method occurs in case of consonant (plosives) prolongation and in any type of repetition. In speech these
events raise a problem of automatic analysis window adaptation. We see automatic analysis window adaptation as a useful technique for discovery of repeated speech patterns with unpredictable length and chaotic occurrence.

5.4 Symbolic Sequence Searching

We created two algorithms [28] (see Fig. 10). One for speech pattern searching (patSearch) and second for searching repeated patterns in speech (repSearch). Last algorithm with help of first pattern searching algorithm, automatically discovers all repeated sequences in a reduced time domain (thanks to SAX representation). In large relational databases, short query set is executed in a large set of data. Our key idea in patSearch is the opposite of the search technique used in case of relational database. We query in a short sequence P a “moving window” of a long sequence C. This opposite approach allows execution of our algorithm (patSearch - parameters: s is a shift, l is C length) in a reduced memory space.

1: while i < n do
2: In i-th window 1st block set to P, remaining blocks put to C.
3: Compute Pos for P. ▶ Pos is a suffix array.
4: With Pos construe Tab for P. ▶ Tab is a look up table.
5: while s < l do
6: Use Tab to query C in P.
7: Save patterns position and patterns length.
8: end while ▶ End: patSearch().
9: end while ▶ End: repSearch().

Key feature of repSearch algorithm is its adaptation capability to unknown repeated speech pattern length. According to its input parameters (8 symbolic speech sequence, w window length, b block length, n number of blocks), it computes proximity measures, which allows to find all occurrences of pattern P in a long symbolic speech sequence C independently of matching length P in C.

5.5 Feature Extraction

During the calculation of the distance between two vectors, we performed standardization. Let X be normal random variable with mean \( \mu \) and variance \( \sigma^2 \). Standardizing X we define a new random variable

\[
Y = \frac{X - \mu}{\sigma}.
\]  

To calculate the speech features (MFCC), we maintain standard method used in Hidden Markov Model Toolkit: Hamming window length (0.025 s), overlapping adjacent frames (0.01 s) and number of bandpass filters (20). Each frame was processed with such attributes to conserve MFCC vector with 13 coefficients. Prior to SAX discrete transformation of speech, we calculated the short time energy over 0.01 s (10 ms) frames (Figure 7). The MFCC coefficients are in the form of matrix. In the case of SAX, we get for the same interval only vector of symbols. Algorithm outputs are always moved by one block (100 ms). We considered these shifts, when the following three derived features of 100 ms blocks were computed:

- **patterns average redundancy** indicates average redundancy in blocks
- **patterns relative frequency** indicates relative frequency of redundancies in blocks

We sum rows of algorithm outputs to have feature **patterns redundancies sum** (left side of the picture, Figure 11). **Patterns average redundancy and patterns relative frequency** is computed along columns of triangular matrix (Figure 11). These features we iteratively calculated for every 5 s long speech interval iteratively shifted by 100 ms.

5.6 Evaluation

In order to compare performance of studied features (algorithms output Specrep and Symrep) to commonly used MFCC coefficients, objective assessment was provided by using feature vectors as inputs for SVM. SVM and their variants and extensions, often called kernel-based methods (or simply kernel methods), have been studied extensively and applied to various pattern classification and function approximation problems [1]. In our experiment, we used sigmoid kernel function, (8). During the process of classifier design, one of intermediate steps is the measurement of data class separability, which we prove with correlation between two classes (fluent speech and dysfluent speech).

In the next stage we studied the data characteristics, for two classes with Mann-Whitney U-test. Nonparametric Mann-Whitney U-tests examines the equality of class medians of random variables \( X, Y \).

\[
U = \frac{n_x (n_x + 1)}{2} - W
\]  

where \( n_x \) is the number of observations in the group and \( W \) is the rank sum statistic related with Wilcoxon’s rank sum test.

Examining correlations and applying Eq. 32 helped us to compare proposed features on basis of MFCC to features on basis of symbolic sequences. Both proposed features and MFCC were used as input for Support Vector Machines with sigmoidal kernel function. We use confusion matrix to measure SVM models’ performance. In addition to accuracy (Eq. 13), we compute sensitivity and specificity of confusion matrix.

\[
\text{Sensitivity} = \frac{TP}{TP + FN} \quad \text{(33)}
\]

\[
\text{Specificity} = \frac{TN}{FP + TN} \quad \text{(34)}
\]

Value of sensitivity (= 1) for a model determines absence of false negative predictions. Analogously to sensitivity, if the model has value of specificity (= 1), the model does not commit any false positive predictions [12].

5.7 Results

Table 7 and Table 8 compare features computed from spectral domain (Specrep - Dynamic Time Warping on basis of MFCC features) and features computed from symbolic sequences (Symrep - our developed algorithms on basis of SAX). Low correlation coefficients \( r \) in Table 7 refer to a fact, that between features computed from fluent intervals and from dysfluent intervals of speech there is a low linear dependence. It is evident, that between our proposed features Specrep average and Symrep sum is the lowest linear dependency.
In case of class separability it is substantial that p-values between Specrep average and Symrep sum are above 0.05. Therefore correlation is not significant, which implies that features are not linearly dependent. According to correlation values, features clearly separate group of fluent and group of dysfluent events.

Results of studied data characteristics of proposed features are shown in Table 8. Nonparametric Mann-Whitney U-tests in Table 8 are significant, where their p-values are below 0.05 level. $h$ values specify accepted hypotheses. We fail to reject hypotheses $h = 0$ only for features Specrep sum and Symrep frequency. Rejected hypotheses $h = 0$ in other features depict that features with $h = 1$ do not have equal medians. According to test results, features with $h = 1$ have unequal data distribution for group of fluent and group of dysfluent features.

We divided data to training (80 %) and testing (20 %) sets. In next step, we trained 6 individual SVM with sigmoidal kernel function. In Table 9 are evaluated the MFCC, Specrep and Symrep features. For MFCC feature we get 50.2 % accuracy. Specrep average achieved 85.4 % accuracy, which is 35.2 % better than base MFCC feature. SVM model for Symrep sum sensitivity do not commit any false negative prediction and produced only 6 % false positive predictions. According to the represented classification results in Table 9, the Symrep sum maintains the upper limit with 97.6 % accuracy.

6. Discussion

Simple dysfluencies (i.e. category of dysfluencies at people with stuttering, known from SLP literature) were processed in many works before. In order to provide basic comparison with our work, we provide typical results of other authors. Authors [35] in 2011 reached 64.01 – 86.19% of accuracy for recognition of syllable repetition (label $R_1$) [42] in 2012 achieved notable 84% of accuracy for the same (i.e. syllable repetition) dysfluent events. For stuttered speech, we were the first to show that SVM with sigmoid kernel function achieved in task recognition of simple dysfluencies (repetition of words and prolongation of a part of word) achieve more accurate results (99.05%) as presented in the published papers (Tab. 1).

Inspired by the video segmentation and key frame extraction (video analysis), transforming time series into symbolic sequences (data mining) and DNA sequence analysis (bioinformatics), we used these technologies in a new application area, the processing of human speech.
### Table 7: Correlations between group of fluent and dysfluent events in proposed features (Source: [28]).

<table>
<thead>
<tr>
<th>Feature</th>
<th>r</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specrep average</td>
<td>0.0356</td>
<td>0.3654</td>
</tr>
<tr>
<td>Specrep frequency</td>
<td>0.3434</td>
<td>0.0</td>
</tr>
<tr>
<td>Specrep sum</td>
<td>-0.0971</td>
<td>0.1032</td>
</tr>
<tr>
<td>Symrep average</td>
<td>-0.0668</td>
<td>0.0</td>
</tr>
<tr>
<td>Symrep frequency</td>
<td>0.1120</td>
<td>0.0</td>
</tr>
<tr>
<td>Symrep sum</td>
<td>-0.0296</td>
<td>0.6205</td>
</tr>
</tbody>
</table>

### Table 8: Mann-Whitney U-test between group of fluent and dysfluent events along examined features (Source: [28]).

<table>
<thead>
<tr>
<th>Feature</th>
<th>h</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specrep average</td>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>Specrep frequency</td>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>Specrep sum</td>
<td>0</td>
<td>0.1659</td>
</tr>
<tr>
<td>Symrep average</td>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>Symrep frequency</td>
<td>0</td>
<td>0.2385</td>
</tr>
<tr>
<td>Symrep sum</td>
<td>1</td>
<td>0.0</td>
</tr>
</tbody>
</table>

We derived our own functions to detect prolongations of stuttered speech.

For previously unsolved problem of detecting complex dysfluencies (i.e. complex repeated events) we have developed new algorithms, operating on the basis of symbolic sequences with using data structure known from DNA analysis. Based on outputs of our algorithms, we derived new features of complex dysfluencies. We used computational intelligence to objectively analyze MFCC, derived MFCC and our new features of complex dysfluencies. We showed that the proposed features of complex dysfluencies bring significant decrease of errors and between compared methods maintain the highest limit 97.6% of accuracy in recognition complex dysfluent events.

Summary overview of the scientific contribution of this work is in Fig. 12. This work is part of research tasks carried out within the doctoral studies at Faculty of Informatics and Information Technologies, Slovak University of Technology (FIIT STU) and at Institute of Informatics, Slovak Academy of Sciences (II SAS).

### 7. Conclusion

Statistical apparatus shows that for searching complex dysfluent phenomena, speech features based on symbolic sequences are competitive in comparison with speech features based on spectral domain.

We developed algorithms designed especially for complex dysfluency detection to solve the problem of dynamic window adaptation in symbolic sequences of speech. Paper shows that speech transformation to discrete symbolic sequences offers an alternative way of speech processing. SAX allows to apply DNA sequence analysis to speech, this potential advantage was enabled by highly efficient data structure, suffix array.

New designated features, capturing phrase repetitions, were statistically analyzed. Objective assessment of new features and MFCC were compared by SVM. SVM established on classical methodology (on terms of simple dysfluencies), trained by MFCC features to recognize complex dysfluencies, have 50.2% accuracy. SVM in case of our features based on symbolic sequences accomplished 97.6% accuracy on identical groups of speech data.

We presented derived functions for prolongation detection, inspired by domain of video analysis. On the basis of our observation that the dysfluencies have recursive property, our novel algorithms were developed. Our technical documentation, beside frequent dysfluencies, pay attention also to dysfluencies which were not studied at all (according to our literature surveys) or are not included (i.e. only rarely categorized) in classical sets of dysfluent symptoms in SLP.

Practical utilization of novel algorithms we see in two different domains: in ASR framework as dysfluency detection module and also in SLP as core of analysis tool for objective assessment of speech therapy.

Presented algorithms may be extended in the future by computing Hamming distance between observed symbolic sequences to obtain higher accuracy. Beside short time energy, we plan to seek speech features in spectral domain to use its symbolic representation to detect also other types of dysfluent events. The algorithms running time (in average 51 times faster than DTW) could be further improved with using suffix tree, which is a similar data structure to suffix array.

Graphical Processing Units (GPU) enable dramatic increases in computing performance at single precision computations, but still have problems with double precision computations (e.g. decrease in performance, support only a few mathematical operations). In future speech analysis by symbolic sequences (e.g. discrete numbers) we propose to exploit the power of GPUs.
Acknowledgements. This contribution was supported by Grant Agency VEGA SR 1/0553/12, 1/0458/13 and supported by Research & Development Operational Programme funded by the ERDF project RPKOM, ITMS 26240220064. We thank for Department of Speech Analysis and Synthesis, Slovak Academy of Sciences for their insightful recommendations about speech processing. Last but not least we thank for Erika Pálfy (Institution for Children with Communication and Learning Disorder, Diálog, s.r.o.), for her help in communication disorder, stuttering.

References


Selected Papers by the Author


