Computing and Informatics, Vol. 29, 2010, 489–501

A COMPARISON OF DECISION TREE CLASSIFIERS FOR AUTOMATIC DIAGNOSIS OF SPEECH RECOGNITION ERRORS

Miloš Cerňak

Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9
845 07 Bratislava, Slovakia
e-mail: milos.cernak@savba.sk

Manuscript received 2 December 2008; revised 21 July 2009
Communicated by Liberios Vokoros

Abstract. Present speech recognition systems are becoming more complex due to technology advances, optimizations and special requirements such as small computation and memory footprints. Proper handling of system failures can be seen as a kind of fault diagnosis. Motivated by the success of decision tree diagnosis in other scientific fields and by their successful application in speech recognition in the last decade, we contribute to the topic mainly in terms of comparison of different types of decision trees. Five styles are examined: CART (testing three different splitting criteria), C4.5, and then Minimum Message Length (MML), strict MML and Bayesian styles decision trees. We apply these techniques to data of computer speech recognition fed by intrinsically variable speech. We conclude that for this task, CART technique outperforms C4.5 in terms of better classification for ASR failures.

Keywords: Decision trees, classification, fault diagnosis, speech recognition

Mathematics Subject Classification 2000: 68T10: Pattern recognition, speech recognition; 62H30: Classification and discrimination; cluster analysis; 62C05: General considerations; 62C12: Empirical decision procedures; empirical Bayes procedures

1 INTRODUCTION

Present automatic speech recognition (ASR) systems are becoming still more and more complex. Technology advances, optimizations, or special requirements such as a small footprint cause that a proper handling of system failures must be established. This process is also called a diagnosis, distinguishing system and service diagnosis. The first is an obvious one, usually solved with detailed logging, aiming at finding a failure block within the system. The second one is more challenging, diagnosing erroneous speech recognition. Here, the diagnosis represents a computing mechanism that looks for error patterns (or error regions) and investigates error source, rather than evaluates the system in terms of some figures of merits.

In this paper we follow the second approach, trying to indentify the sources of faulty speech recognition service. We use well-known approach using decision trees [1]. Here, classifiers are trained to classify some kind of speech recognition error. Out-of-vocabulary, acoustic or language model errors, belong to the classic outcome of error classification used in the speech recognition. In speech recognition evaluation, the standard metric called word error rate is used, which is based on three kind of errors: insertions (ASR wrongly inserts a word), deletions (ASR deletes correct word) and substitutions (ASR substitutes correct word with wrong word). In our work the outcome of the classification (the classified event) is the kind of error done: insertion, deletion, or substitution. If the classifiers generalize enough, we can use them also for prediction of the classified event. All the information used in the classification is called predictors. As in speech community different kinds of classifiers were used, we present a comparison of these classifiers on the classification task described above. The paper is focused on fair comparison of the classified methods on the same task. As a by-product we present the results of the diagnosis for the best classifier. Based on these results we can further improve modeling of the error source.

The diagnosis process can be modeled as a conditional probability $P(W|X)$ of a word sequence W given by a set X of measured predictors. We can claim that well specified set of predictors leads to a good estimation of the confusion measure $CM(W) = P(W|X)$ as well, the related task of diagnosis. As the number of used predictors is usually high, selection of most informative predictors is done using another computing mechanism. Classifiers trained from well- and mis-recognized examples are used here. In principle any kind of classifiers may be used (such as neural networks, SVMs), but in the last decade, failure diagnosis or diagnostic evaluation of speech recognition using decision trees has become more popular. Most of the approaches use either the CART method [1, 2, 3] or the C4.5 algorithm [4, 5]. CART stands for "Classification And Regression Trees", and is a well known decision tree induction method with its root in the statistics community [6]. On the contrary, the family of C4 decision tree learning systems, with C4.5 as the one of the latest releases [7], originated from the machine learning community. Regarding fault diagnosis, these methods are quite similar as both try to predict faults with the ultimate goal to find possible fault sources. The advantage of this approach for

diagnosis is that a lot of speech recognition results can be processed automatically. Although the decision tree diagnosis approaches mentioned above do not belong to the best classifiers, they benefit from outputs readable for human supervisors.

This paper contributes to the comparison of decision tree classifiers for speech recognition. Several decision tree methods are examined. The diagnosis is based on ASR experiments that use intrinsically variable speech, which include fast, slow, loud, soft (low, not whispered) speech, plus questioning and normal style speech.

The paper is structured as follows: The following Section 2 introduces decision tree classifiers in test. Section 3 describes used data and ASR experiments, on which decision tree (DT) diagnosis is applied. Experiments and comparisons are described in Section 4. Finally, Section 5 discusses the results.

2 DECISION TREE CLASSIFIERS

Decision trees are classifiers that represent their classification knowledge in tree form (usually in binary tree form). Each interior node of a decision tree is a test on an attribute. Satisfying that test causes the instance being classified to take one branch out of that node, failing the test causes the instance to take the other branch. A decision tree is used to classify an instance by starting at the root node of the decision tree and following the path the attribute tests dictate until a leaf node is encountered. Each leaf node in a decision tree is a decision, i.e., represents a classification. An instance that ends up at some particular leaf node is classified with the class assigned to that leaf node. A second kind of tree is a class probability tree. This has a vector of class probabilities at each leaf instead of a decision [8].

The basic algorithm builds a tree top down using the standard greedy search principle, based on recursive partitioning. The partitioning algorithm includes stopping, splitting and pruning rules.

2.1 Classification and Regression Trees

Most of the automatic failure diagnosis use either CART or the C4.5 method (see e.g. [9, 10]). We can classify three most popular CART styles that differ in splitting rule:

- CART style using Gini index of diversity [8]. Gini looks for the largest class in the training list and strives to isolate it from all other classes. It produces good results for a large variety of classification problems and is thus the default rule used for CART. Gini index of diversity minimizes the risk involves when making predictions once having made the test, using the following equation:

$$\begin{aligned} G(class|test) &= \sum_{i=1}^{T} Pr(outcome\, i) G(class|outcome\, i) = \\ &= \sum_{i=1}^{T} \frac{n_{i,\cdot}}{n_{\cdot,\cdot}} \sum_{j=1}^{C} \frac{n_{i,j}}{n_{i,\cdot}} \left(1 - \frac{n_{i,j}}{n_{i,\cdot}}\right). \end{aligned} \tag{1}$$

Here $n_{i,j}$ corresponds to the number of examples at the node being evaluated that fall in test outcome i and have class j, $n_{.,j}$ is the number that has class j regardless of test outcome, and $n_{i,.}$ is the number that has test outcome i regardless of class. T is total number of tests, and C is total number of classes.

- CART style using information gain (entropy) splitting rule. The Entropy rule, which is very similar to twoing in practice, strives for similar splits. The split is motivated by minimization of entropy between a parent node and a sum of entropies of two child nodes. The criterion maximizes the information gained about the class making the test. The following formula is used [8]:

$$\begin{aligned} I(class|test) &= \sum_{i=1}^{T} Pr(outcome\, i) I(class|outcome\, i) = \\ &= -\sum_{i=1}^{T} \frac{n_{i,.}}{n_{.,.}} \sum_{j=1}^{C} \frac{n_{i,j}}{n_{i,.}} \log_2 \frac{n_{i,j}}{n_{i,.}}. \end{aligned} \tag{2}$$

- CART style using "twoing". The philosophy of twoing is far different from that of Gini. Rather than initially pulling out a single class, twoing first segments the classes into two groups, attempting to find groups that together add up to 50 percent of the data. Twoing then searches for a split to separate the two subgroups. The twoing rule strikes a balance between purity and creating roughly equal-sized nodes.

2.2 C4.5 Method

C4.5 style uses Quinlan's gain ratio splitting rule [7]. Information Gain measure in terms of Equation (2) can be defined as

$$\begin{aligned} Gain(class|test) &= I(test) - E(class|test) = \\ &= -\sum_{j=1}^{C} \frac{n_{i,j}}{n_{i,.}} \log_2 \frac{n_{i,j}}{n_{i,.}} + \sum_{i=1}^{T} \frac{n_{i,.}}{n_{.,.}} \sum_{j=1}^{C} \frac{n_{i,j}}{n_{i,.}} \log_2 \frac{n_{i,j}}{n_{i,.}} \end{aligned} \tag{3}$$

where $I(test)$ measures randomness of the distribution of examples under test over C possible classes, and $E(class|test)$ is expected information for the tree with *class* as root.

The Quinlan's modification consists of dividing $Gain(class|test)$ by the following expression

$$IV(class) = \sum_{j=1}^{C} \frac{n_{i,j}}{n_{i,.}} \log_2 \frac{n_{i,j}}{n_{i,.}} \tag{4}$$

obtaining the Gain Ratio

$$Gain_R(class|test) = \frac{I(test) - E(class|test)}{IV(class)}. \tag{5}$$

According to Quinlan

> the rationale behind this is that as much as possible of the information provided by determining the value of an attribute should be useful for classification purpose.

2.3 Minimum Encoding Styles

Minimum encoding approaches were developed for fitting models to data problems. The problem of finding a good model is converted to a problem of finding minimum encoding of the data, using concepts from Shanoons theory of information. Minimum encoding styles can also be considered as extensions of decision trees, which may result in decision graphs. They are based on minimum description length principle and minimum message length principle (MDL/MML). These principles use "encoding length" to measure the quality of hypotheses. We examined three styles as defined and implemented by the IND program: strict MML (SMML) that is closely related to the theory, a modified approach MML which does not penalize large tree as strongly, and *Bayesian trees*. The theoretical background is described in [11, 12], and their application in ASR diagnosis task can be found in [13].

3 ASR EXPERIMENTS

3.1 OLLO Database

The speech database used for ASR experiments is the Oldenburg Logatome Corpus (OLLO) [14]. It contains 150 different non-sense utterances (logatomes) spoken by 40 German and 10 French speakers. Each logatome consists of a combination of consonant-vowel-consonant (CVC) or vowel-consonant-vowel (VCV) with the outer phonemes being identical.

A large drop in recognition accuracy in ASR is not only encountered in noisy environments, but even when clean speech in known conditions is to be recognized. This drop is often caused by speech intrinsic variabilities (as for example speaking rate, style or effort, speaker's age, gender or health condition, regional dialect or accent).

To provide an insight into the influence of speech intrinsic variabilities on speech recognition, OLLO covers several variabilities such as speaking rate and effort, dialect, accent and speaking style (statement and question). The OLLO database is freely available at `http://sirius.physik.uni-oldenburg.de`.

Each of the 150 logatomes was recorded in six variabilities (speaking rate: fast and slow; speaking effort: loud and soft; speaking style: spoken as question and normal) with three repetitions. This results in 2,700 logatomes per speaker. Influences caused by dialect may be investigated, as speakers without dialect and from four different dialect/accent regions were recorded. Utterances of ten speakers with

no accented speech (five speakers for training and five speakers for testing) were selected for ASR tests.

3.2 Automatic Speech Recognition Test Setup

Hidden Markov Models (HMM) and Gaussian Mixture Models (GMM) based speech recognition system is trained using public domain machine-learning library TORCH [15] on the training set that consists of 13 446 logatome utterances. Three states left-right HMM models were trained for each of the 26 phonemes in the OLLO database including silence as well. Gaussian mixture models with diagonal covariance matrices were used to model the emission probability densities of the 39 dimensional feature vectors – 13 cepstral coefficients and their derivatives (Δs) and double derivatives ($\Delta\Delta s$). All the features were calculated using the HTK hcopy tool. We calculated MFCC vectors every 10 msec using windows of 25 msec size. We found experimentally that GMM with 17 Gaussians per state perform the best, so we used this setup in further tests. The phoneme HMMs are connected with no skip. We extended the TORCH library in a package of calculation and storage of feature data, necessary for further statistical processing. The decoder collects the feature data by running on the testing set that consists of 13 466 logatome utterances. We trained and tested the ASR system with MFCC feature set.

We recognized single logatome PCMs with no grammar, doing in fact phone based recognition using phonotactic model. The key idea behind was to eliminate the influence of language model (LM) constrains to the recognition process, focusing more on acoustical information given by the signal. Average phoneme recognition accuracy of the ASR systems on this task was 76.49 % (see Table 1). Similar performance can be achieved also by the HTK tool, but we preferred using TORCH as it is much easier to adapt speech recognition process.

Accuracy	Deletions	Insertions	Substitutions
76.49 %	3.60 %	3.69 %	16.23 %

Table 1. Phone recognition accuracy of ASR system on logatom recognition task

4 EXPERIMENTS AND COMPARISONS

More than 13 K logatomes were used for training of acoustic models (AM), and the same amount of different data from a test set was used for ASR testing. The test set was further split into 90 % for development (DT training), and 10 % for evaluation.

The input waveform was always a single logatome from the test set, but we performed phone recognition as decribed in the previous section. We extended the TORCH library to new dumping of all the data used later in DT analysis. Along with each logatome recognition, error information was recorded (if there was insertion, deletion or substitution), and a following vector was created (the first item was a classified item (error information), others were predictors' values X):

- Error category (was the classified item)
  - deletion (values 0, 1; 1 for deletion made)
  - insertion (values 0, 1; 1 for insertion made)
  - substitution (values 0, 1; 1 for substitution made)
- Speech variability:
  - V1: Fast speech
  - V2: Slow speech
  - V3: Loud speech
  - V4: Soft, low (not whispered) speech
  - V5: Questioning style speech
  - V6: Normal speech
- Logatome type (VCV or CVC)
- Speaker ID (from S06 to S10)
- Speaker gender (M, F)
- Speaker age (ranges 21–31 and 32–42 years old)

CART style trees were generated using several splitting rules (see Section 2.1), subsetting on multivalued discrete variables, cost-complexity pruning and 10-fold cross validation [6]. C4.5 style trees were generated using Quinlan's gain ratio splitting rule (see Section 2.2), and Quinlan's pruning rule [7]. The depth of the inducted trees was not limited. For each of the error category, the following decision trees were trained and examined:

1. C4 style using Quinlan's gain ratio splitting rule
2. CART style using Gini index of diversity
3. CART style using information gain (entropy) splitting rule
4. CART style using "twoing"
5. a Bayesian tree
6. a MML tree
7. and a strict MML tree.

The IND program [8] was used for DT training and testing. For each DT style, we trained three DTs, for deletions, insertions and substitutions, 7×3 trees in total. Figure 2 shows an example of a trained tree. For each of the trees, misclassification matrix was calculated. Table 2 shows an example of a strict MML matrix calculated from substitution data. Our criterion for best tree selection was the best error prediction ([2,2] elements of the matrices: the misclassification predicted vs. actual error). The lower the missclassification rate is, the better classifier (predictor) of

	Actual not subst. (0)	Actual subst. (1)	Total
Subst. not predicted (0)	0.537147	0.312036	0.849183
Subst. predicted (1)	0.060178	0.090639	0.150817
	0.597325	0.090639	1.000000

Table 2. Misclassification matrix of the SMML style DT for substitutions (predicted – row values vs. acutal – column values)

the error made we have. Having the best tree, we examined also a path leading to the most probable terminal node with the error prediction.

As substitutions were done more often than deletions and insertions (see Table 1), our primary interest was in diagnosis of this kind or errors. Figure 1 shows the misclassification rates of all the substitutions classifiers. Bayesian and SMML styles DTs are the best according to the criterion used.

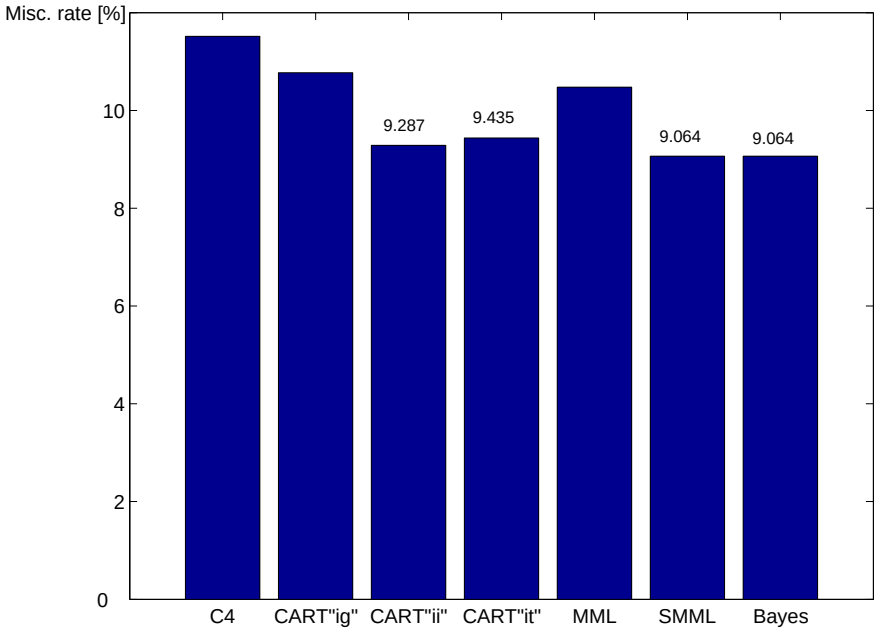

Fig. 1. Misclassification rates of the substitution classifiers. CART styles splitting rules abb.: "ig" is Gini index of diversity, "ii" is information gain, and "it" is twoing.

Table 3 presents the results we got from the trees examining the path leading to the most probable terminal node with the error prediction. The order of found predictors on the path is not important.

Error category	Selected DT	Selected predictors
Deletions	CART using entropy	Fast speech and S06 speaker (more probable) Fast speech and rest of speakers (less probable)
Insertions	Strict MML	VCV logatoms Slow speech (more probable) Loud speech and female speakers (less probable) Questioning style and female speakers
Substitutions	Strict MML	Male (or age 32–42), CVC type, soft speech Female (or age 21–31), CVC, loud and soft

Table 3. Results for each of the error category

4.1 Deletions

For deletions, the best tree was the CART style using information gain (entropy) splitting rule. The best predictor for deletions made was specified as fast speech. All the rest of instrinsic speech variabilities were contributing less to this error category. The second major predictor selected was speaker identification (S06): male gender with speaker age between 32 and 42 years. The second minor predictor were the rest of speakers, mostly (3/4) female recordings, all 21–31 years old.

4.2 Insertions

For insertions and substitutions, strict MML DTs were the best trees. Analyzing insertions, primary error predictors were selected as VCV type of logatomes and slow speech. The minor predictors were selected as loud and questioning style speech together with female recordings.

4.3 Substitutions

The hardest specification of predictors was in case of an analysis of substitutions. There were no clear patterns, also overall classification errors of the trees were less than for DTs for insertions and deletions. Anyway, CVC type of logatomes and soft speech were in most cases dominant predictors of substitution prediction.

5 CONCLUSIONS AND FUTURE WORK

We contributed to the topic of ASR evaluation, focusing on classical error categories and intrinsic speech variabilities. We specified predictors for different DT styles which contribute to the each error category, which is a novel approach in ASR diagnosis.

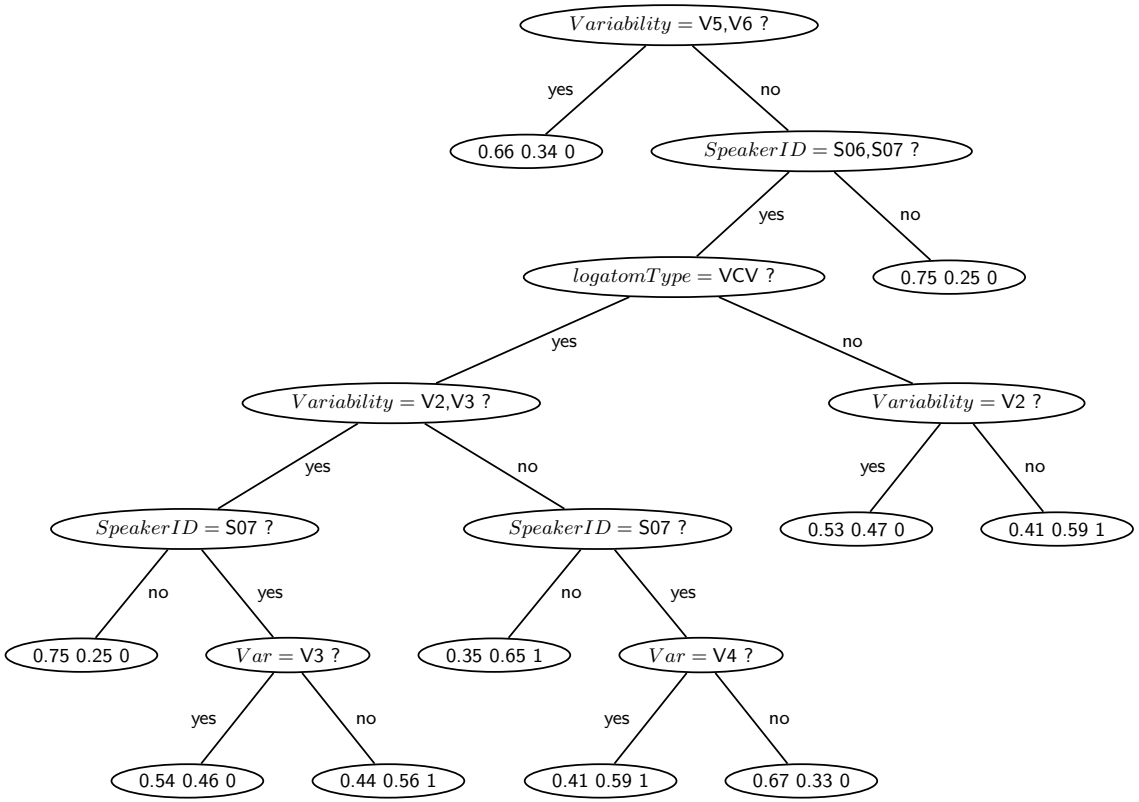

Fig. 2. A CART class probability tree trained using information gain splitting rule for the substitution diagnosis task. Each terminal leaf contains a vector of class probabilities for "0" (a substitution is not predicted) and "1" (a substitution is predicted), accompanied with the final decision.

We compared 5 binary decision tree styles for 3 different error categories. We conclude that

1. CART classifiers for the task outperform C4 classifiers.
2. The best CART DT style is information gain (entropy) splitting rule (for deletions category the misclassification rate was 0, as the only DT, the comparison for substitutions can be seen in Figure 1).
3. We recommend to use and further investigate minimum encoding styles DTs, as they perform almost equally to deletions CART DTs, and even slightly outperform substitutions CART DTs.

Often the quality of a predictive model depends on insightful predictor creation. Subject-matter expertise is critical. In future, we want to look for some data-driven mechanism to decrease our dependency on such expertise. However, until that time, we will look for new powerful predictors of ASR errors.

Concerning generalization of our results to diffrent tasks in speech recognition, we emphasize that our comparison was made on the OLLO vocabulary. The vocabulary consisted of CVC isolated words in comparable quantity as used by Steeneken in his diagnostic work [16]. We believe that the results could be generalized also for the different, more complicated tasks for speech recognition, such as large vocabulary speech recognition tasks [17].

In the spirit of making research reproducible [18], all the train and test data, including scripts and description how the trees were built and used is freely available at `http://www.ui.savba.sk/speech/milos_web_data/ollo_dt_comparison.tar.gz`. Only the IND program has to be downloaded and installed. The IND program is freely available at `http://opensource.arc.nasa.gov/software/ind/`. To our knowledge it is the only free tool that offers such broad functionality for decision tree testing. It was tested on several universities as well, and we supppose that its influence on the accuracy and the results presented is minimal.

Acknowledgments

The work presented in the paper was supported by Slovak Research and Development Agency under research project APVV-0369-07, and with support from the VEGA grant No. 2/0138/08.

REFERENCES

[1] Chase, L.: Error-Responsive Feedback Mechanisms for Speech Recognizers. Ph. D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 15213, USA, April 1997.

[2] Peters, S. D.—Stubley, P.—Valin, J. M.: On the Limits of Speech Recognition in Noise. In ICASSP '99, March 1999, Vol. 1, pp. 365–368.

[3] Cerňak, M.—Wellekens, C.: Diagnostics of Speech Recognition Using Classification Phoneme Diagnostic Trees. In Proc. of Computational Intelligence 2006 (Special Session on NLP), November 2006, San Francisco, CA, USA.

[4] Greenberg, G.—Chang, S.: Linguistic Dissection of Switchboard-Corpus Automatic Speech Recognition Systems. In Proc. of ITRW on Automatic Speech Recognition: Challenges for the new Millenium, Paris, France, 2000, pp. 195–202.

[5] Zheng, A. X.—Lloyd, J.—Brewer, E.: Failure Diagnosis Using Decision Trees. In ICAC '04: Proceedings of the First International Conference on Autonomic Computing (ICAC '04), Washington, DC, USA, 2004, pp. 36–43, IEEE Computer Society.

[6] Breiman, L.—Friedman, J.—Stone, Ch. J.—Olshen, R. A.: Classification and Regression Trees. Chapman & Hall/CRC, New York 1983.

[7] Quinlan, R. J.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Meteo, CA, USA, 1993.

[8] Buntine, W.: Tree Classification Software. Technology 2002: The Third National Technology Conference and Exposition, Baltimore, USA, December 1992.

[9] Zhou, G.—Deisher, M. E.—Sharma, S.: Causal Analysis of Speech Recognition Failure in Adverse Environments. In ICASSP '02, Vol. 4, pp. 3816–3819.

[10] Hunt, M.: Speech Recognition, Syllabification and Statistical Phonetics. In Proc. of ICSLP, Jeju Island, Korea, October 2004.

[11] Wallace, C. S.: Statistical and Inductive Inference by Minimum Message Length. Springer-Verlag, Information Sci. & Stats., May 2005.

[12] Buntine, W.: A Theory of Learning Classification Rules. Ph. D. thesis, University of Technology, Syndey 1991.

[13] Cerňak, M.—Darjaa, S.: Noisy Speech Recognition Failure Diagnosis Using Minimum Message Length Decision Trees. In Proceedings of the 15th International Conference on Systems, Signals and Image Processing (IWSSIP), June 25–28, 2008, Bratislava, Slovak Republic.

[14] Wesker, T.—Meyer, B.—Wagener, K.—Anemuller, A.—Mertins, A.—Kollmeier, A. B.: Oldenburg Logatome Speech Corpus (OLLO) for Speech Recognition Experiments with Humans and Machines. In Interspeech 2005, September 2005, pp. 1273–1276.

[15] Collobert, R.—Bengio, S.—Marithoz, J.: Torch: A Modular Machine Learning Software Library. Technical Report IDIAP-RR 02-46, IDIAP 2002.

[16] Steeneken, H. J. M.—Varga, A.: Assessment for Automatic Speech Recognition: I. Comparison of Assessment Methods. Speech Communication, Vol. 12, 1993, No. 3, pp. 241–246.

[17] Juhár, J.—Rusko, M. et al.: Development of Slovak GALAXY/VoiceXML Based Spoken Language Dialogue System to Retrieve Information from the Internet. Interspeech – ICSLP, Pittsburgh, USA, 2006, pp. 485–488.

[18] Vandewalle, P.—Kovačević, J.—Vetterli, M.: Reproducible Research in Signal Processing: What, Why, and How. IEEE Signal Processing Magazine, Vol. 37, May 2009.

Miloš Cerňak studied telecommunication engineering at Slovak Technical University in Bratislava, where he received the Bachelor degree (1999) and a Master degree (2001) both in informatics, and a Ph. D. degree (2005) in telecommunication engineering. During his eight years carrier he got Boeing scholarship at Iowa State University in USA, and during the last four years he worked in basic and applied research in France (Eurecom Institute), Belgium (Acapela Group) and Czech Republic (IBM). He leads a project focused on speech recognition systems diagnosis. He is a member of IEEE and Slovak Acoustical Society; he serves as ad-hoc reviewer of IEEE Transactions on Speech and Audio Processing, and Elsevier Signal Processing.