TWO-STEP RUSSIAN PHONEME CLASSIFICATION METHOD

A. M. Soroka, P. D. Kukharchik

Belarusian State University
Minsk, Belarus
E-mail: soroka.a.m@gmail.com, p.d.kukharchik@gmail.com

Key words: phoneme based speech recognition, support vector machine, wavelet transform

INTRODUCTION

The main approach of speech recognition is based on the classification of basic speech units, such as phonemes or diphones. However, standard feature extraction methods make such speech units very irregularly distributed in the feature space. There are several methods that address feature vector clusterization, such as the construction of language lattices and tangling grids or various linguistic models [1]. These methods have some drawback, with algorithms complexity among others.

In this paper two feature extraction methods based on wavelet transform are proposed. They attempt to make feature vectors more evenly distributed in the feature space. We also propose a genuine two-step phoneme classification method based on support vector machine (SVM), which deals better with clustered phonemes than an ordinary multiclass SVM classifier.

The structure of the paper is as follows: first, we describe the main idea of the SVM algorithm; then, we introduce two wavelet transform based feature vector construction methods; after that we present the two-step phoneme classification method; and finally, we describe the some experiments, which measure the performance of all the proposed methods.

SUPPORT VECTOR MACHINE

SVM is a supervised learning algorithm, which was first proposed by Vapnik [2]. This algorithm attempts to minimize empirical risk by utilizing a set of separating hyperplanes, which not only separate different classes of data, but also maximize margins between the classes. This fact made SVM highly generalization-capable. Let's consider the SVM method in details.

Let X be a set of d-dimensional vectors ($X \subseteq R^d$) and $A \subset X$ be it's subset. Consider the following mapping: $f : X \to Y$, where $Y = \{-1, +1\}$, $f(x) = +1$ if and only if $x \in A$, and $f(x) = -1$ otherwise. This mapping defines two distinct classes.

The mapping itself (as well as underlying A) is considered to be unknown, and the goal of the algorithm is to learn to classify elements of X into the two classes (in other words, to imitate f).

The learning process is based on a finite set of training instances $\tilde{x} \in \tilde{X} \subset X$, for which values $f(\tilde{x})$ are considered known. The combination of all the instances and the appropriate values of f (the set of tuples $(\tilde{x}, f(\tilde{x}))$) is called the *training set.*

We will also use the term *"training class"* to indicate a set of training instances that belong to one of the two classes defined by f.

In the simplest case the training data are linearly separable. That is, there exist $w \in R^d$ and $b \in R$, such that $\forall \tilde{x} \in \tilde{X}$: $f(\tilde{x})(w \cdot \tilde{x} - b) \geq 0$ (1). Values of w and b define a hyperplane $w \cdot x = b$ that separates the training classes. An infinite number of hyperplanes may satisfy (1). From the point of view of statistical learning theory [2] only one of the hyperplanes is of a particular interest. It is called the *optimal separating hyperplane;* it introduces the maximum possible margin between the training classes.

The training data are not always linearly separable, i.e. it is not always possible to find a separating hyperplane. In practice the dataset is often disturbed by noise and some of the training instances may be bad representatives of their class. So it is better to allow these instances to be "misclassified." Such kind of misclassification is handled by attaching a penalty $\xi : \tilde{X} \to R$ for each training instance. The sum of these penalties is added to the cost function. Having found the optimal separating hyperplane, one can now define a linear step classifier, which would differentiate between the two classes $c(x) = sign(w \cdot x - b)$

Attaching penalties is not the only way of dealing with linear non-separability in SVM. Sometimes the linear non-separability of the dataset is caused not only by noise but also by "internal structure" of the data. To build a non-linear classifier, first, set X is mapped into some Hilbert space F (called the *transformed feature space*), which can be of a much higher dimension than X. Then the linear separation process described above is performed on the elements of the transformed feature space. To solve optimization problem in the feature space one doesn't even need to know the explicit form of the mapping function $\varphi : X \to F$. The only thing required for the calculations is the so called *reproducing kernel* of the mapping (or simply the kernel). This kernel merely defines the inner product of the transformed feature space in terms of elements of set X (i.e. original feature space), i. e. $k : X \times X \to R$, $k(x, x') = \langle \varphi(x), \varphi(x') \rangle_F$. It can be shown [7] that kernel function uniquely identifies mapping φ and vice versa.

The resulting step classifier would look as follows: $c(x) = sign(k(w, x) - b)$.

Choosing the appropriate kernel is quite an involved process, which lies beyond the scope of the current paper.

WAVELET TRANSFORM BASED CONSTRUCTION OF FEATURE VECTOR

One of the basic aspects of acoustic signal processing is the process of feature vector construction. The utilization of mel-frequency cepstral coefficients (MFCCs) [5] is one of commonly used methods used for that purpose. It was shown, however, that this method lacks proper accuracy due to unacceptable proximity of the resultant feature vectors in the feature space [6].

In our paper we propose alternative method, which is based on wavelet transform of the acoustic samples. The method obtains feature vector in the following way. First, the sample's wavelet is split into $3N$ windows. Next, in each window an average energy is calculated: S_{ij}, $i=1...N$, $j=1...3$. The S_{i2} are put directly into feature vector, while S_{i1} and S_{i3} are differenced first: $\Delta_i = S_{i3} - S_{i1}$, which is done to deal with several effects induced by vowel reduction and coarticulation at the beginning and at the end of phonemes.

The resultant feature vector is constructed as follows:

$$x = \left(S_{12}, \ \ldots \ S_{N2}, \ \Delta_1, \ \ldots \ \Delta_N\right).$$

The method is illustrated on fig. 1.

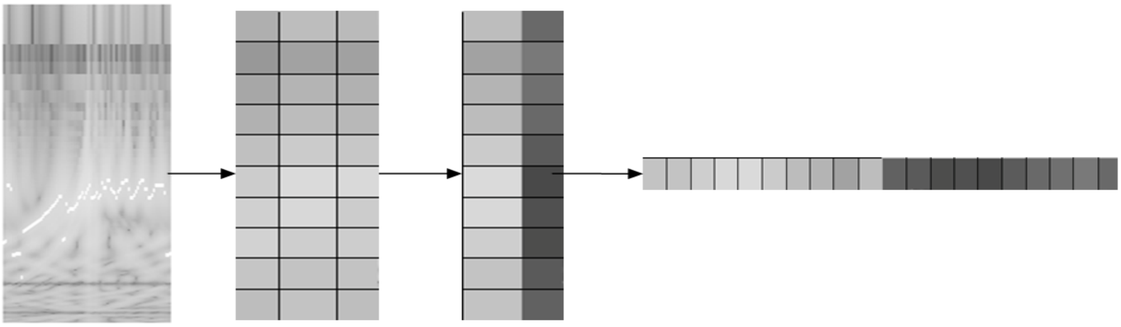

Fig. 1. The process of obtaining feature vector from the sample's wavelet

TWO-STEP PHONEME CLASSIFICATION METHOD

Now we will describe the two-step phoneme classification method. The method attempts to determine the phoneme, to which a given acoustic sample may correspond to.

As one could infer from the method's title, the method constitutes of two steps. The first step determines a group of phonemes, to which a given sample would most likely correspond to. After that, the second step determines the exact phoneme that the sample most probably represents.

The first step utilizes a so called multiclass classifier, which is simply a set of binary SVM classifiers, each having been trained to classify into a separate phoneme group.

The second step also utilizes a set of binary SVN classifiers. However, there are two major distinctions from the first step: first, the set of classifiers is determined by the phoneme group chosen during the first step; and second, each classifier is trained to classify the instances of one phoneme against the instances of some other phoneme. In other words, for each possible pair of phonemes in the chosen phoneme group a classifier is built to discriminate between the phonemes within the pair. Such approach proved to increase the classification accuracy.

The phoneme groups were formed empirically to minimize classification errors during the first step. First, we considered all the phonemes to be separate groups, i.e. each phoneme group contained only one phoneme (a so called *single-step classifier*). Having build the set of classifiers needed for the first step, we run several tests and found several pairs of phoneme groups, one member of which was often misclassified to be the other one (e.g. sound samples that represented phoneme "Ш" were often interpreted by the classifier to be the representatives of phoneme "Щ" and vice versa). Then, we merged most frequently misclassified group pairs. Having reduced the classification error during the first step (as well as the amount of phoneme groups), we reiterated the process again to reduce the classification error even more, until it became low enough.

EXPERIMENTS

To build and test the two-step classifier a training set of 7000 sound samples obtained from VoxForge speech corpus [4] and the acoustic base of the department of radio physics and digital media technologies of the Belarusian State University was used. The training set contained about 150 samples for each of the Russian phonemes. A test set of 1000 samples was used. The proposed method of feature vector construction was tested as well as the standard MFCC-based method.

During the first experiment a training set of 1000 samples, 700 of which were the representatives of phoneme "A", and a test set of 300 phoneme "A" samples was used. The determined accuracies of the classification of the proposed method and MFCC were 82,7%, and 80,3%, respectively.

To test the classification ability of similar phonemes we used a training set of 1000 vowel phoneme samples, with the test set being 100 samples of phoneme "A". During this test the two method performed with the following accuracies: 92% (the proposed method) and 82% (MFCC).

During the third experiment we attempted to assess the performance of the optimal two-step classifier. The parameters of the optimal classifier were determined via cross-validation and grid search. It utilized the proposed feature vector construction method.

Having built the optimal two-step classifier we tested it's performance with the training set of about 6000 samples (about 135 samples for every phoneme) and a test set of 1000 samples, which contained only the samples representing "A", "M", "H", or "Д". The results of the third test are summarized in table 1.

Table 1

The phoneme classification accuracy test results

	А	М	Н	Д
Accuracy of the first step, %	98,8	92,0	93,6	92,0
Accuracy of the second step, %	90,0	93,2	90,4	93,6

The fourth experiment compared the proposed two-step classifier with simple single-step classifier (i.e. the classifier, for which the phoneme groups contained only one phoneme each). The results of the experiment are shown in table 2.

Table 2

Single-step classifier vs. two-step classifier

	А	М	Н	Д
Single-step classifier accuracy, %	84,8	79,2	76,0	77,2
Two-step classifier accuracy, %	89,2	85,2	83,6	84,8

The data presented in table 2 shows that two-step classifier outperforms the single-step classifier on 6,4% on average.

CONCLUSION

In this paper we have considered feature vector construction method based on wavelet transform. The method proved to outperform the traditional MFCC method by 2,4% on average and by 10% during the classification of closely spaced phonemes.

We also proposed two-step phoneme classification method based on SVM, which outperforms an ordinary multiclass SVM classifier by 6,4% on average.

REFERENCES

1. *Алиев, Р. М.* Поиск ключевых слов с использованием решетки фрагментов слов / Р. М. Алиев, Цзинбинь Янь, И. Э. Хейдоров.// Компьютерная лингвистика и интеллектуальные технологии : сб. материалов ежегод. междунар. конф. «Диалог 2009», Бекасово, 27–31 мая 2009 г. / Рос. фонд фундам. исслед., Моск. гос. ун-т ; редкол.: А.Е. Кибрик [и др.]. М., 2009. С. 351–354
2. *Vapnik, V.* The nature of statistical learning theory [M] // New York. Springer-Verlag, 1995
3. *Шмырев, Н. В.* Свободные речевые базы данных VoxForge.org // Сборник трудов международной конференции «Диалог 2008». 2008. С. 585—588.
4. *Huang, X.* Spoken Language Processing: a guide to theory, algorithm, and system development. / X. Huang, A. Acero // New Jersey: Prentice-Hall Inc. Upper Saddle River, 2001.
5. *Siafarikas, M.* Speech Recognition using Wavelet Packet Features / M. Siafarikas, I. Mporas, T.Ganchev, N. Fakotakis. // Journal of Wavelet Theory and Applications. 2008. Vol. 2, № 1. P. 41—59.
6. *Bart, Hamers*. Kernel models for large scale applications // Katholieke Universiteit Leuven – Faculteit Toegepaste Wetenschappen Arenbergkasteel, B-3001 Heverlee (Belgium)