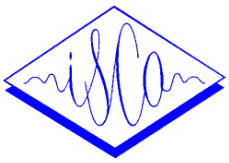

2nd European Conference on
Speech Communication and Technology
EUROSPEECH '91
Genova, Italy, September 24-26, 1991

DVQ : DYNAMIC VECTOR QUANTIZATION APPLICATION TO SPEECH PROCESSING

Franck Poirier

Télécom Paris
Département Signal
46 rue Barrault
75634 Paris Cedex 13

ABSTRACT

In this paper, we present a modified version of the Learning Vector Quantization (LVQ) called Dynamic Vector Quantization (DVQ). We compare the performances of both classifiers based on competitive learning on speech classification tasks.

All the experiments clearly highlight the ability of generalization of the DVQ algorithm, it gives best result than LVQ2 on the test set. Moreover, DVQ always provides substancial gain in memory size and consequently is less time-consuming.

Keywords : classification, vector quantization, evaluation, artificial neural network, acoustic-phonetic decoding, speech recognition.

1-INTRODUCTION

Artificial Neural Networks (ANN) are alternate tools for classification tasks. They have a great generalization ability and well work for many real-world problems such as acoustic-phonetic decoding in a speech recognition system.

ANN paradigm can be described as non-parametric model that learn some kind of internal representation of their inputs from quite a few examples. In other words, ANN finds the hidden structure of the problem. This is why, if we want to extract a discriminant representation of the speech pattern, we have to study the learning and generalization abilities of ANNs, even if actually HMMs perform better.

In this paper we present and evaluate a new version of the LVQ algorithm called Dynamic Vector Quantization (DVQ). DVQ proceeds in an incremental way, by starting the training phase with only one reference vector per class and progessively increasing the number of references.

Two corpus have been used. The first one contains multi-speaker isolated letters and the second one consists of phonetically balanced sentences.

2- SPEECH DATABASES

2-1 Multispeaker database

The corpus is composed of the French alphabet plus the phoneme /e/. The speech database is a subset of BDSONS with 28 speakers who pronounced 4 times the 27 words of the corpus. The speech database contains 3024 letters, 18 speakers (9 male and 9 female) are used for training (1944 letters) and 10 other speakers (5 male and female) are used for testing (1080 letters).

The speech waveform is sampled at 8kHz. Every 10ms, using 20ms overlapping Hamming window, an 8-dimension MFCC (Mel Frequency Cepstrum Coefficients) vector is computed.

For such a specific vocabulary, it is possible to use acoustic and phonetic knowledge in order to reduce the amount of data and to automatically retain discriminant input for the classifier [1].

In our experiment, each letter is described by only three discrinant events such as beginning or end of vowel, frication before or after the vowel, plosion, voiced occlusion, vowel nucleus. Each event is associated with one frame of 20ms. Thus, each letter is represented by a 24 (3*8) dimensional vector.

It can be noticed that the recognition task of spoken letters by several speakers is difficult. The main reasons are the inter-speaker variability, the acoustic closeness of some letters (A and K, J and G, M and N, P and T ...), the short duration of these words and the variety of phonemes (25 phonemes on 33 in French) .

2-2 Phonetically balanced sentences

The corpus is composed of 200 french phonetically balanced sentences uttered naturally by a male speaker. The corpus contains 5270 phonemes, each one was labelled by hand at the center. The corpus is halfed between a learning set and a test set.

The sufficient consistency of the data was checked, using k-Nearest Neighbours classifiers [2].

The speech waveform is sampled at 10kHz. Every 10ms, using 33ms overlapping Hamming window, an 8-dimension MFCC vector is computed. Each phoneme is represented by 7 frames, 3 frames on the left and 3 frames on the right of the label, which correspond to 56 coefficients (7*8).

It can be noticed that a phonetically balanced corpus is not well fitted for learning. In fact, some phonemes are very little frequent (/ɲ/, /g/, /ɥ/ ...), for that reason it is far from obvious that they are correctly learnt. So, it would be better to modify the learning set in order to increase the occurence of some phonemes. A minimal number of those in the learning set is required to adequately learn some of their references .

3- CLASSIFIERS

Different classifiers (nearest neighbours, SOFM, LBG, LVQ2, DVQ) have been used on speech data. Usually, they are all used as static pattern classifiers or vector quantizers. It is self-evident that in a recognition system they must be preceded and followed by other modules (segmentation, temporal normalization, lexical decision ...).

For the isolated letters, a reference system based on DTW (Dynamic Time Warping) has been also used.

Nearest neighbours and LBG are well-known; we have already presented the use of SOFM in [3]. So, only LVQ and DVQ are described in the following.

3-1 LVQ

LVQ is similar in structure to the feature map classifier but LVQ is a supervised Nearest Neighbour classifier [4]. So, training data must be labelled.

LVQ is a vector quantizer which gives a codebook of k reference vectors. Contrary to SOFM, the codebook has no topology.

LVQ-codebook vectors are defined near the optimal decision borders between the classes in the sense of Bayesian decision theory. So, very good practical results has been obtained with it on synthetic data [3]. LVQ-codebook has a highly discriminant ability.

LVQ has already been been used for some speech recognition tasks [5, 6, 7].

LVQ2, the most popular version, is pictured in Figure 1. It adjusts slighty the reference vectors in a direction towards the input vector x that attempts to improve classification performance.

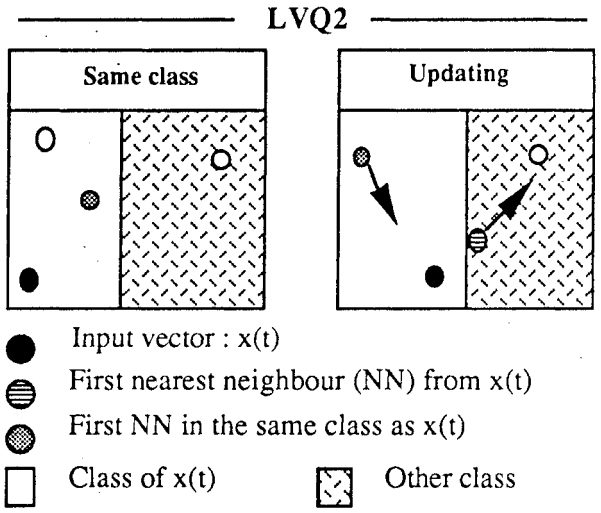

Fig. 1. LVQ2

Here, we want to underline the drawbacks of the LVQ2 algorithm.

First, the number of reference vectors is constant. Number and initial position must be decided at the beginning. So, the initialization step is crucial. SOFM or k-means algorithm can be used for the initialization of the codebook. In a way, the frontier complexity is fixed.

Second, if LVQ2 is applied for a too long time, two phenomenons can occur. If the classification task is not so hard, the codebook becomes too specialized which involves a poor ability of generalization on the test set. In the other case, if the task is too hard the reference vectors drift away and go out of the input pattern space [8].

Third, in order to initialize the reference vectors or to increase the ability of generalization, it is advised to perform pre-processings (SOFM, k-means or editing algorithms) before LVQ2. Accordingly, pre-processing plus LVQ2 are very

time consuming and difficult to tune together. The pre-processing phase is often much longer than the adaptation phase.

3-2 Dynamic Vector Quantization

We present now a new version of the LVQ algorithm called Dynamic Vector Quantization (DVQ). DVQ allows to simplify the initialization procedure, to decrease the number of codebook vectors and, we hope, to improve the ability of generalization. Moreover, DVQ is well fitted to complex data like speech data (various density functions).

Our goal is to adapt dynamically the number of codebook vectors to the intrinsic complexity of the training set and more precisely to the complexity of each class. Instead of fixing the same number of reference vectors per class, DVQ initializes the codebook with only one reference vector per class (center of gravity). During the adaptation phase, the algorithm acts as follow (cf. Figure 2) :

if

the closest reference from the input vector x does not belong to the class of x

and

the closest reference in the class of x is too far from x (distance greater than σ)

then

create a new reference equal to x

else

use the LVQ2 rule.

The threshold σ is a function of the minimum class variance; σ can be a constant or function of time.

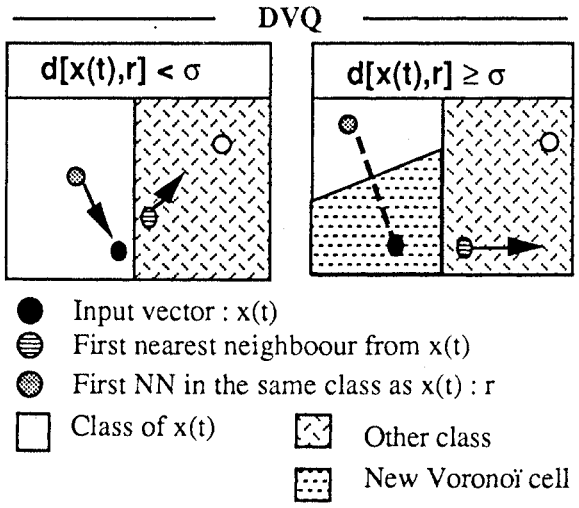

Fig. 2. DVQ

4- EXPERIMENTS AND RESULTS

4-1 Results for the isolated letters

Table 1 shown the recognition rate for male and female speakers on the training and test sets for all the experiments.

With LVQ2, on the training set, the recognition rate is 98% but it decreases, on the test set, at 71% for male speakers and 67% for female. LVQ2 creates too precise frontiers which induces bad generalization. In order to decrease the precision of the frontiers, the training set must be smoothed by the editing algorithm (some letters are removed from the set by using the k-nearest neighbours rule). In this case, the recognition rate is 72% on the test set for both male and female speakers. However, the whole procedure is very time consuming.

DVQ performs better than other classifiers. Compared with LVQ2, DVQ shows an improvement of about 5%. For male speakers the recognition rate is 92% on the training set and 77% on the test set. Moreover, pre-processing (editing algorithm) has no effect and is useless.

A reference word recognition system based on DTW is worst when it uses all the signal frames and is much more time-consuming.

Method	Speakers	Editing algorithm	Learning Data	Test Data
DTW	female	no	100	70.5
	male	no	100	70.5
LVQ2	female	no	98	67
	male	no	98	71
	female	yes	95	72
	male	yes	91	72
DVQ	female	no	93	74
	male	no	92	77

Table 1. Recognition rates on isolated letters

4-2 Results on phonemes

Tables 2 to 4 shown the recognition rate on the training and test sets.

For all the experiments (same number of references for both methods), DVQ is slightly better than LVQ2.

On 27 phonemes (10 vowels and 17 consonants, 2348 examples in the learning set), DVQ starts with 27 references and stops with 100. For the same number of references, LVQ2 performs 2% worst. When we use the editing algorithm as a pre-processing method, obviously the recognition

rate on the learning set is very high but the ability of generalization is very poor (the recognition rate falling from 96% to 66%). In fact, this algorithm only reduces the error rate for the set on which it has been applied (learning set).

Those disappointed results may be explained by the fact that the energy coefficient or the temporal derivative of the MFCC are not used. So a number of errors occurs between consonants and vowels. It is more interesting to classify separately vowels and consonants.

On 10 vowels (1074 examples in the learning set), DVQ requires only 38 references and shows an improvement of about 4% (recognition rate of 80% on the test set). As this task is less difficult than the previous one, the editing algorithm does not make the results on the test set worst.

On 17 consonants (1274 examples in the learning set), the differences between both algorithms are very slight (1%). However, DVQ has the main benefit to automatically find the adapted number of references.

Method	Number of Ref.	Editing	Learning Data	Test Data
LVQ2	**100**	**no**	**83**	**70**
	100	yes	96	66
DVQ	**100**	**no**	**85**	**72**
	45	yes	98	69

Table 2. Recognition rates on 27 phonemes

Method	Number of Ref.	Editing	Learning Data	Test Data
LVQ2	100	no	98	82
	36	**no**	**94**	**76**
	100	yes	98	82
	36	yes	98	75
DVQ	**38**	**no**	**94**	**80**
	33	yes	100	81

Table 3. Recognition rates on 10 vowels

Method	Number of Ref.	Editing	Learning Data	Test Data
LVQ2	100	no	97	77
	64	**no**	**93**	**76**
	100	yes	98	75
	64	yes	99	73
DVQ	**60**	**no**	**93**	**77**
	34	yes	100	77

Table 4. Recognition rates on 17 consonants

6- CONCLUSION

For all these experiments, DVQ provides substancial gain in memory size, speed and more often performance. We show that DVQ is well fitted to complex data like speech data classification (various density functions).

Asymptotically, DVQ should perform as well as LVQ2. As a matter of fact, DVQ performs better for the following reasons :

- at the beginning all the phonemes are equally represented (one reference per phoneme),
- during the learning phase, the number of references per phoneme dynamically increases function of the complexity of each class,
- the final number of references is not a priori decided.

Further researches will be focused on the integration of DVQ with other classifiers to improve the performances on speech data.

Acknowledgement

We would like to thank F. Bimbot who made available to us the phonetically-balanced sentences database.

References

[1] F. Poirier, "Knowledge-based Segmentation and Feature Maps for Speech Recognition". ICSLP 90, Kobe, 90.

[2] F. Bimbot et al., "Phonetic Features Extraction using Time-Delay Neural Networks". ICSLP 90, Kobe, 90.

[3] F. Poirier, "Improving the training speed and the ability of generalization in learning vector quantization: DVQ". ICASSP'91, Toronto, 91.

[4] T. Kohonen, "Self-organization and associative memory". Springer Verlag, 88.

[5] S. Makino et al., "A Japanese text Dictation System Based on Phoneme recognition using a Modified LVQ2 Method". ICSLP 90, Kobe, 90.

[6] P. Ramesh et al., "A new connected word recognition algorithm based on HMM/LVQ classification". ICASSP '91, Toronto, 91.

[7] Y. Bennani et al., "Validation of Neural Net Architectures on Speech Recognition Tasks". ICASSP '91, Toronto, 91.

[8] T. Kohonen, "The Self-Organizing Map". Proc. of the IEEE, vol. 78, september 90.