

On the Optimization of Multiclass Support Vector Machines Dedicated to Speech Recognition

Freha Mezzoudj and Assia Benyettou

Laboratory Signal-IMage-PArole (SIMPA), Department of Computer Science,
University of Science and Technology of Oran USTO, Algeria, 31036
{freha.mezzoudj,a_benyettou}@yahoo.fr

Abstract. We present in this paper an interesting approach to enhance the performance of multi-classification using Genetic Algorithm. Two systems for an instance selection and feature selection are respectively introduced. We combined Genetic Algorithm with multiclass Support Vector Machines in order to reduce the learning set. The goal is to simplify the learning process and to improve the generalization. The results obtained on speech corpus show encouraging improvements in terms of processing time and classification accuracies.

Keywords: Support Vector Machines, Genetic Algorithms, Multi-classification, Speech recognition, Machine learning, Feature Selection, Instance Selection.

1 Introduction

Realizing an Automatic Speech Recognition system (ASR), requires a parameterization step such as Cepstral analysis on the Mel scale (MFCC) of the speech signal in order to get acoustic vectors. The information contained in these vectors should be explored to decide to which phonetic classes do they belong. A learning step is necessary to build reference models for all considered phonemes. The recognition of unknown information (test) is realized with similar measure of this new signal with regards to the learning models.

The Support Vectors Machines (SVM) introduced by Vapnik [1] is a technique of classification based on the statistical learning theory. During the last years SVM have proved satisfactory even excellent performance in various fields [2, 3]. In general, the SVM provide a unique solution, because the learning phase is based on solving a quadratic problem (QP). The support vectors (SVs) are the most informative samples used in the definition of the class boundaries.

In many machine learning applications, the used learning dataset may contain hundred or thousand of instances, which are described with numerous features. In some situations, reduced set of samples can be sufficient to interpret the results. Most features in high dimensional vectors may usually be noisy and redundant. They seriously affect the generalization performance and make the learning process slow and difficult. It is desirable to reduce the SVM learning time for complex applications such as speech recognition.

In literature, reduction methods regarding SVM revolve around two main axes: the first idea is to optimize the SVs obtained after the learning by eliminating the less

important with minimal impact on accuracy [4]. The second idea is to reduce the training data size or dimensionality before the learning phase [5]. We are interested by the second one.

This paper is organized as follows: in section 2, we briefly sketch the ideas behind SVM for classification. In section 3, we present the principle to use Genetic Algorithms for instance and feature selection. We noted our hybrid systems: GA-SVM-IS (Instance Selection for SVM by GA) and SVM-GA-FS (Feature Selection for SVM by GA). In the experimental section, we do a phonetic multi-classification with a fast, but not very popular SVM: Carmmer and Singer method. We use dataset from TIMIT corpus, with two different parameterizations of speech signal. Then we apply an instance selection on the learning set. We also propose a feature selection. In this step, we extract straight blocks of MFCC in order to protect the speech signal. Finally, some conclusions and further work are presented.

2 Support Vector Machines

The SVM based on the principle of Structure Risk always find a global solution. They leave the largest possible margin on either sides of the decision boundary. In general, Classification aims to build an efficient and effective model on the training set to predict class labels of unknown data. Formally, let the training data: $A_n = \{(x_i, y_i), i = 1..n\}$, $x_i \in \mathfrak{R}^m$ and class labels $y_i \in \{-1, +1\}$. The input x_i is a vector that we call a “pattern” or “instance” of m components which we call “features”. In our case, the instances correspond to phonemes and the features are values of MFCC. The goal of a supervised learning algorithm is to find the decision function $f(x)$ of the separating hyperplane written as: $w \cdot x + b = 0$, where w called “weight vector” is normal to the hyperplane and the parameter b leads to a translation of the separating hyperplane from the origin.

We consider two canonical hyperplanes, where the parameters w, b are constrained by: $\min_i |w \cdot x_i + b| = 1$. We take two samples x_1 and x_2 from the canonical hyperplanes H_1 and H_2 , which are parallel and equidistant to the separating hyperplane, satisfying respectively equations: $(\langle w \cdot x_1 \rangle + b) = +1$ (for $y = +1$) and $(\langle w \cdot x_2 \rangle + b) = -1$ (for $y = -1$). Thus the margin γ measured perpendicular to the optimal hyperplane $H(w, b)$ is the distance between the two samples so: $\gamma = 2/\|w\|$.

The margin value is proportionally inversed to the $\|w\|$, to maximize the margin implies minimizing $\|w\|^2$, so the learning task in SVM can be formalized as the following quadratic optimization problem (QP):

$$\text{Min } \left\{ \frac{1}{2} \|w\|^2 \right\} \quad (1), \quad \text{Subject to the constraints: } y_i (\langle w \cdot x_i \rangle + b) = 1 \quad (2)$$

As a result of solving the QP in (1)-(2), Lagrange multipliers α_i are usually used to incorporate the restrictions into the functional, converting it into a dual form, and applying a quadratic programming technique (we refer readers to [1, 6] for a full description of the technique), we obtain the final SVM classifier :

$\sum_{i=1}^{n_{sv}} \alpha_i^* y_i \langle v_i \cdot x_i \rangle + b^* = 0$, where n_{sv} is the number of SVs (v_i), and α_i^* are the nonzero Lagrange multipliers.

The basic SVM can be extended to support the noisy data. The proper classification conditions (2) are reformulated by incorporating slack variables " ξ_k " which relax the constraints and the parameter "C" that controls the permitted error.

SVM are originally designed for binary classification problems, henceforth two approaches have been proposed to extend them to multiclass case. There is an indirect approach, which consists of subdividing the original multiclass problem into a collection of binary-class sub-problems such as One-vs-One (OVO) and One-vs-Rest (OVR) methods. The second approach is directly considering all data in one optimization formulation such as Crammer and Singer (CS) methods [7]. Our proposed approach works with any of these multiclass methods, even if in our experiments we have focused on Crammer and Singer (CS) SVM multiclass described in [8] because we have found it faster than OVO and OVR methods. Even Huang et al. [9] experimented various techniques of Multiclass SVMs with the different parameters, and found that the CS method is optimal one.

3 Genetic Algorithms for Instance and Feature Selections

Instance selection (IS) is the technique that selects an appropriate reduced subset of the training samples. It is considered as the dual problem of feature selection [10]. In our case, it reduces the possibility of selecting noisy training samples as SVs.

Feature selection (FS) allows the identification of the most relevant features for the classification task. Feature selection methods are classified into three types: Filters, Embedded methods and Wrapper [10]. The approach for feature subset selection proposed in this paper is an instance of the wrapper methods. These methods use the predictive accuracy of predetermined classification algorithms, as criteria to determine the goodness of a subset of features

Genetic algorithms are specifically designed to resolve problems involving large search spaces containing multiple local minima. They are search algorithms based on the mechanics of natural selection. They are not simple random walk, but they efficiently exploit historical information to speculate on new search points with expected improved performance [11, 12]. We used them to solve IS and FS problems. In addition, some FS and IS methods with GA were proposed in the literatures [5, 13, 14]. It is quite natural to design the solution of these two problems as a chromosome which is the vector that simulates all features or all instances. The fitness function is the reduction rate achieved between the selected instances (or features) and the whole data set. Also, the main steps of our two proposed recognition speech systems based on multiclass SVM and GA noted GA-SVM-IS and GA-SVM-FS are as follows:

Program1 GA-SVM-IS (Naïve implementation)

```

1: Inputs:
   .Training samples Xtrain=[x1,...xn], Class labels Ytrain,
   Test examples Xtest, Ytest;
   .SVM parameters: C, kernel, nbclass;
   .GA parameters:{NIND:Number of chromosomes per sub-
   populations, MAXGEN: Number of generations, NVAR=Chrom.
   Length = n = Xtrain instances Number, Pcrossover,
   Pmutation=1/Chromosome Length, etc};
2: Create initial population{NIND Chromosomes of NVAR
length + intelligent chromosome of 1}and Evaluate it
with SVMmulticlass;
3: Repeat until gen <= MAXGEN
4: Create population:{NIND Chromosomes of NVAR length};
   % Evaluation of population:
5: For each chromosome, i=1:NIND
6:   For each instance, j=1:NVAR
7:     If Chrom(i,j)==1 then
8:       NewXtrain ← instance(j)from Xtrain
9:   End for
   % Apply the classifier:
10: α=SVMmulticlass Learn(NewXtrain,NewYtrain);
11: (Yprededction)= SVMmulticlass Classify (Xtest,Ytest);
12: Compute the fitness: error(Yprededction vs Ytest);
13: End for
14: Best = best chromosome in current population;
15: Apply Genetic operators: Crossover, Mutation,
   offspring Reinsertion into current population;
16: gen = gen+1;
17: End Repeat
18: Output
   Instances ranked, Recognition rate=100-error.

```

The principle of GA-SVM-FS is similar to GA-SVM-IS, and only differences were noted.

Program2 GA-SVM-FS (Naïve implementation)

```

%....
NVAR=Chrom. Length =  $m'$  = Maximum Number of MFCC blocks;
%....
%Creation and evaluation of population;
5: For each chromosome, i=1:NIND
6:   For each instance, j=1:n
7:     For each set-feature (block MFCC)
8:       If Chrom(i,j)==1

```

```

9:   then NewFeatures= [NewFeatures, a block of MFCC];
10:   End for
11: NewXtrain ← NewFeatures(j) from Xtrain
12: End for
13: Apply the classifier
14: End for
%....

```

4 Datasets and Software

We applied our systems on a corpus of 18 classes (6 Vowels, 6 Fricatives and 6 Plosives) from the DARPA TIMIT acoustic phonetic speech corpus. The speech signals were sampled at 16 kHz cepstral analysis on the Mel scale which is commonly used in speech recognition, taken every 20 ms a Hamming window of 25 ms. The parameterization was based on 12 Mel Frequency Cepstral Coefficients and their corresponding residual energy. We added the energy because it is less affected by the transmission channel than the spectral characteristics. So the phonetic vectors obtained are blocks of 13 MFCC which correspond to 416 feature in maximum.

We have used the same corpus where the first and the second derivatives Delta and Delta-Delta coefficient through HTK are added [3]. Dynamic cepstral features are shown to play an essential role in capturing the transitional characteristics of the speech signal [3]. The phonetic vectors are blocks of 39 MFCC which correspond to 1677 features in maximum.

Table 1. Description of 18 phonemes Corpus with 13 MFCC

Class	Phoneme	Train	Test
Vowels	/ah/	2200	879
	/aw/	700	216
	/ax/	3352	1323
	/ax-h/	281	95
	/uh/	502	221
	/uw/	536	170
Fricatives	/dh/	2058	822
	/f/	2093	911
	/sh/	2144	796
	/v/	1872	707
	/z/	3574	1273
	/zh/	146	74
Plosives	/b/	399	182
	/d/	1371	526
	/g/	1337	546
	/p/	2056	779
	/q/	3307	1191
	/t/	3586	1344
Total	instances	31514	12055

Table 2. Description of 18 phonemes Corpus with 39 MFCC

Class	Phoneme	Train	Test
Vowels	/ah/	2195	879
	/aw/	699	216
	/ax/	3446	1346
	/ax-h/	361	118
	/uh/	502	221
	/uw/	536	170
Fricatives	/dh/	2688	1053
	/f/	2093	912
	/sh/	2138	796
	/v/	1883	710
	/z/	3571	1273
	/zh/	146	74
Plosives	/b/	2067	886
	/d/	3376	1245
	/g/	1929	755
	/p/	2475	957
	/q/	3435	1244
	/t/	4154	1535
Total	instances	37694	14390

We employed, the CS SVM multiclass of Joachims proposed in [8].

5 Results and Discussion

For TIMIT data corpus (13MFCC), we retained the value of C and the kernel proposed in [4]. For the GA's parameters, we took : Generation number=50, Chromosome length (GA-SVM-IS): $n=31514$, Chromosome length (GA-SVM-FS): $m'=416/13=32$.

After applying GA-SVM-IS system, we noted that nearly the half of the original training set is selected 15684 instances. Using GA-SVM-FS system, 16 vs 32 blocks in maximum (so 208 values) are selected.

Table 3. Comparison of classification for corpus with 13 MFCC

System	SVM- CS	GA-SVM-IS	GA-SVM-FS
Recognition Rate (%)	58.30	59.24	58.42
Time-train (sec.)	66.55	37.20	55.83
Time-test (sec.)	0.19	0.22	0.22
#SV	42	40	36
Training set size	(31514,416)	(15684 , 416)	(31514, 208)

We proposed to make a classification by phonetic class. We classified the six vowels, then the fricatives, and finally the plosives. We recapitulated the results in table 4.

For the feature selection, we noticed that vowels which are the longest phonemes contain much redundancy than fricatives. For the plosives most MFCC blocks are informative.

Table 4. Classification for TIMIT corpus with 13 MFCC by phonetic class

System	SVM-CS	GA-SVM-IS	GA-SVM-FS
Recognition Rate (%)	71.74	72.90	72.18
Time-train (sec.)	09.80	05.39	11.38
	15.49	05.09	09.22
	11.59	04.02	07.42
Time-test (sec.)	00.03	00.01	00.05
	00.01	00.03	00.03
	00.05	00.02	00.05
Training set size	(7571, 416)	(15684 , 416)	(7571, 286)
	(11887, 299)	(5931 , 299)	(11887, 195)
	(12056, 286)	(6063 , 286)	(12056, 221)

For TIMIT data corpus with 39 MFCC, we retained the same training parameters for SVM and GA of the precedent experiment, but the Chromosome length for GA-SVM-IS is $n = 37694$ and for GA-SVM-FS is $m' = 1677/39 = 43$. We considered the classification by phonetic class too (see Table 6).

Table 5. Classification for corpus with 39 MFCC

System	SVM Multiclass	GA-SVM-IS	GA-SVM-FS
Recognition Rate (%)	73.02	76.22	74.62
Time-train (sec.)	1156.36	1156.36	704.57
Time-test (sec.)	0.53	0.25	0.27
#SV	281	195	176
Training set size	(37694, 1677)	(19789 , 1677)	(37694, 1053)

Table 6. Comparison of classification by phonetic class for corpus (39 MFCC)

System	SVM Multiclass	GA-SVM-IS	GA-SVM-FS
Recognition Rate (%)	80.00	80.96	80.22
Time-train (sec.)	712.30	542.20	602.50
	239.86	193.56	213.46
	328.03	286.04	315.84
Time-test (sec.)	0.08	0.05	0.05
	0.01	0.04	0.03
	0.16	0.09	0.05
Training set size	(7739, 1677)	(4267 , 1677)	(7739, 1521)
(# instances, # features)	(12519, 1287)	(6951 , 1287)	(12519, 1092)
	(17436, 1209)	(9086 , 1209)	(17436, 1131)

6 Conclusion

In this paper, we use two integrated instance and feature selection schemes based on Genetic Algorithms and Multiclass Support Vector Machines for phonemes classification. We employ them separately to accelerate the learning phase and to improve generalization. The use of the GA in this context allows SVM to achieve better results at competition time. Therefore our systems guarantee good generalization.

In this study, we notice that the improvement achieved by the instance selection is greater than that achieved by the feature selection. This result may be relative to the nature of the SVM classifier and the complexity order of corresponding QP.

The appreciation of the data quality in adding dynamic aspect to the MFCC vectors gives better results in classification.

There are several avenues for this study. The GA parameters need to be carefully chosen to allow the algorithm to converge to good solutions. In alternative work we will use meta-GA for choosing them. We would like to use other versions of GA such as Hybrid Taguchi GA with other classifiers for speech recognition.

References

1. Vapnik, V.: The Nature of Statistical Learning Theory, Berlin, Germany (1995)
2. Wang, L.: Support Vector Machines: Theory and Applications. Springer, Heidelberg (2005)
3. Zribi, B.S., Ben Ayed, M.D., Ellouze, N.: Support Vector Machines Approaches and Its Application to Speaker Identification. In: IEEE International Conference on Digital Eco-Systems and Technologies, DEST 2009, Turkey, pp. 662–667 (2009)
4. Tang, B., Mazzoni, D.: Multiclass Reduced-Set Support Vector Machines. In: 23rd International Conference on ML, USA (2006)
5. Ahn, H., Kim, K.: Bankruptcy Prediction Modeling with Hybrid Case-Based Reasoning and Genetic Algorithms Approach. Appl. Soft Comput. 9, 599–607 (2009)
6. Cherkassky, V., Mulier, F.: Learning from Data: Concepts, Theory and Methods, 2nd edn. IEEE Press (2007)
7. Crammer, K., Singer, Y.: On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. J. Mach. Learn. Research 2, 265–292 (2001)
8. Joachims, T.: SVM^{multiclass} V2.12, http://download.joachims.org/svm_multiclass/
9. Huang, Z., Chen, H., Hsu, C.J., Chen, W.H., Wu, S.: Credit Rating Analysis with Support Vector Machines and Neural Networks: A Market Comparative Study. Decis. Support Syst. 37, 543–558 (2004)
10. Guyon, I., Elissee, A.: An Introduction to Variable and Feature Selection. J. Mach. Learn. Research 3, 1157–1182 (2003)
11. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, Massachusetts (1989)
12. Steeb, W.H., Hardy, Y., Stoop, R.: Genetic Algorithms in The Nonlinear Workbook, 3rd edn. World Scientific Publishing (2005)
13. Boehm, O., Hardoon, D.R., Manevitz, L.M.: Classifying Cognitive States of Brain Activity via One Class Neural Networks with Feature Selection by Genetic Algorithms. Int. J. Mach. Learn. Cybern. 2(3), 125–134 (2011)
14. Nair, S.S.K., Subba, R.N.V., Hareesha, K.S.: An Evaluation of Feature Selection Approaches in Finding Amyloidogenic Regions in Protein Sequences. Int. J. Comput. Appl. 8(2) (2010)
15. HTK: Hidden Markov Model Toolkit: Speech Recognition Research Toolkit, <http://htk.eng.cam.ac.uk/>