AUDIO CONTEXT CLASSIFICATION ON MOBILE DEVICES USING REDUCED COMPLEXITY SVM

Daniele Battaglino[*‡], *Annamaria Mesaros*[†], *Ludovick Lepauloux*[*], *Laurent Pilati*[*] *and Nicholas Evans*[‡]

[*] NXP Software
Valbonne, France

[†] Department of Signal Processing
Tampere University of Technology
Tampere, Finland

[‡] Multimedia Department
Institut EURECOM
Biot, France

ABSTRACT

The environmental contextualization in mobile devices is a key-point for automatically adapting the mobile configuration to different situations. Audio context recognition suits this scope, but has limitations in terms of complexity and memory dedicated, especially in a always-listening mode. We propose a set of techniques to reduce the complexity of Support Vector Machines in terms of number of Support Vectors. Our hypothesis is that a large part of the training samples generates redundant information for the classifier. We investigate the effect of Linear Discriminant Analysis and clustering for training data selection, resulting in a reduced size model, with an acceptable loss in the classification accuracy.

Index Terms— Audio Context Recognition, mobile devices contextualization, SVM, k-means, LDA

1. INTRODUCTION

Audio Context Recognition (ACR) is a task which aims to recognize and categorize the audio information of a scene. In this work we use the term *context* to refer to an ensemble of sounds, events and background noise which identify a particular situation. As expressed in [1], the context concerns automatic classification of an environment around a device: for instance a bus, an office or a street. The motivation for that, as presented in [2], is the continuous demand for advanced functionality with a simple interaction mechanism. In that sense, knowing the context is a fundamental key for making the devices adapt transparently to the situation: changing the ring tone depending if the user is at the office or in a car is an example of that.

The choice of audio (instead of other sensors available on handset device such as light sensors, gyroscopes, accelerometers, etc.), is driven by the breadth of applications: every device has a microphone, while we cannot assume the presence of other sensors. In addition to that, the use of audio has been showed to outperform use of accelerometer measurements, and to add complementary information [2, 3]. The classification algorithm should run directly on the device, in *always-on* mode, as opposed to a *cloud* solution, because a constant internet connection would be too battery consuming. Moreover, the personification of the device according to the situation should not rely on the network presence which cannot be guaranteed in every situation.

Approaches to ACR involve different classification methods, such as distance-based methods, Gaussian mixture models (GMM), Hidden Markov models (HMM), support vector machines (SVM). For example in [4] authors used k-nearest neighbors (kNN) to classify examples based on distance to known data. Another approach uses temporal modeling, such as HMM to classify the context through a sequence of events or states [5]. In this study we use SVM for classification, as it was shown to have the best trade-off between high performance and low-complexity [6].

For good generalization performance, a large amount of data is required to train a good model for classification. However, we have to deal with device limitations, in particular memory size dedicated for the model and the classification complexity. These aspects add constraints and new challenges to perform ACR on a device: on the one hand we want to have a light and fast classification algorithm, and on the other the accuracy loss has to be minimum. SVM training complexity heavily depends on the number of training samples and the classification complexity on the amount of support vectors (SVs) stored in the model. While the training is not an issue from a device point of view (since computed *off-line*), the *on-device* classification is affected by the number of SVs: the complexity is $O(nS)$, with n samples and S support vectors [7].

Our main hypothesis is that in a large data set there is a lot of redundancy. Reduction of training complexity has been approached previously by removing training samples that are not relevant for building the decision boundary [8]. Similar data selection was performed also in [9], where the unnecessary samples far from boundary were pruned. Other approaches introduce the reduction directly during the classification phase, where a small random subset is extracted and used to extract the best margin [10]. These methods result in faster training, but do not attempt to reduce the number of support vectors.

We propose data selection by clustering the training set,

and uniformly selecting training points from all regions, such that the training set size is reduced but keeps the variability of original data. Reduction of the training set size results in a reduction of SVs in the model, and a reduction of performance in classification. We analyze the trade-off between reducing the number of SVs and loss in performance, in the specific case of audio context recognition. The novel contributions of this work to ACR classification are the reduction of features vector dimensions through Linear Discriminant Analysis (LDA) and the reduction on the amount of SVs with clustering selection.

2. METHODS

The methods used for reduction of complexity consist of a set of techniques designed to reduce the number of SVs with the common goal of decreasing the memory size and the computational complexity of the testing phase. Before training, we perform feature extraction and selection, followed by reduction of the training dataset. In testing, the feature selection transformation is applied to the test data before the label prediction. The steps of the system are presented in Fig. 1.

2.1. Features extraction and selection

The first step of the classification system consists of feature extraction. We extract Mel-frequency Cepstral Coefficients (MFCCs) from audio and use their long-term average. In many ACR systems, the MFCC are the reference features [11–13], because they encode the spectrum of the signal in a compact and uncorrelated way. Longer audio clips are divided into non-overlapping segments as investigated in [14]. This is done essentially to deal with different file lengths, resulting in sub-clips of equal length. For each of them we calculate the mean and standard deviation of the MFCC over all frames. Each such sub-clip is represented by a single feature vector, describing its spectrum.

We use Linear Discriminant Analysis to further reduce the size of this feature vector. Linear Discriminant Analysis is a technique that takes into account the label information for finding the best projections. The original dimensional features are projected into a new space, where the ratio of *between-classes* variability to *within- classes* variability is maximized. Knowing the class, we build a new cost function:

$$J(\vec{w}) = \frac{\vec{w^T} S_b \vec{w}}{\vec{w^T} S_w \vec{w}} \tag{1}$$

where S_b is the *between scatter matrix* and S_w is the *within scatter matrix*, calculated in the conventional manner [15]. It is demonstrated that this equation can be written as a regular eigenvalue problem, where we can find eigenvalues and eigenvectors which correspond to the best discriminant features transformations [16]. LDA projection is extracted from the training samples and applied to the test samples before

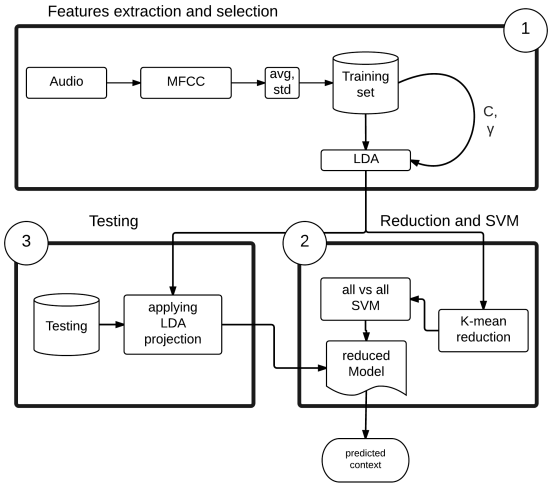

Fig. 1. The entire process of complexity reduction: **1.** feature extraction and selection using LDA. **2.** The SVM training, after the K-means dataset reduction **3.** The testing with SVM reduced model.

classification. The use of LDA does not change the average classification accuracy, but by reducing the dimension of the feature vectors for training SVM, it reduces the size of the model in the memory.

2.2. Reduction and SVM training

The second step consists of the SVM training using only selected samples of the full training set. To select training points uniformly from the full set, we use clustering. For a given context, we have n training data points $x_i, i = 1...n$ that we want to partition into k clusters. K-means is a clustering method that aims to find the positions of the means $\mu_i, i = 1...k$ of the clusters that minimize the distance from the data points to the cluster. K-means clustering solves

$$\arg\min_{c} \sum_{i=i}^{k} \sum_{x \in c_i} d(x, \mu_i) \tag{2}$$

where c_i is the set of points that belong to cluster i. The k-means clustering uses the square of the Euclidean distance $d(x, \mu_i) = ||x - \mu_i||^2$. The best grouping is obtained using an iterative algorithm: after the initialization of the clusters centroids, each data point is attributed to the closest cluster. Then a new mean is calculated from all data points belonging to that cluster. The algorithm is repeated until convergence.

In our system, k-means is applied for each class separately. The cluster centroids are initialized randomly and the dataset is partitioned into clusters. After this, a number of points are selected randomly from each cluster: this way the selected points cover all the original space. During the reduction phase, the cluster centroids are included into the training

set because they represent the core of the training dataset distribution.

The clustering and selection of data points is done for each context, and then the SVM classifier is trained with the reduced training samples. With a smaller training set, the resulting model relies on a smaller number of SVs in determining the separating hyperplanes. This effectively results in a smaller size model that needs to be kept in the memory of the device.

2.3. Testing

The third step of the system is the testing phase. Given a test audio sample, the feature extraction is applied to obtain a test point that will be classified. The MFCCs are extracted from the test sub-clip in the same way described in Step 1, and the sub-clip is represented by a single features vector. Then, LDA projection is applied to reduce the feature vector dimension. Finally the test sample is classified according to the support vectors stored in the model.

3. EXPERIMENTS

The proposed method was tested using two different audio context databases, using a five-fold controlled partitioning into training and testing sets. The results are presented as the average performance of the five folds, context-wise and overall average. The baseline performance is calculated as the classification performance with no reduction of the training set. For a complete evaluation of the proposed dataset reduction method, we also present classification performance for a system with training set reduction using only random selection, i.e. selecting training data points randomly from the training set, without clustering.

3.1. Databases

The DCASE challenge dataset [17] is a small size environmental dataset, consisting of 10 contexts. For each context there are 10 recordings, each of length 30 seconds. For the purpose of SVs reduction, the DCASE dataset is too small in terms of number of training samples. For this reason, a more exhaustive database was collected in NXP Software with the aim to represent better the variability of environmental audio.

The NXP Software database was recorded by volunteers using mobile devices. An application for recording was installed on the devices, asking the users to label the context in which the audio is recorded. Then, the recording was uploaded to the server. The recorded data covers five of the most common everyday audio contexts: inside a bus, inside a car, office, subway and street. The amount of data available for each context is presented in Table 1.

Context	Files	Minutes	Sub-clips
bus	22	121	1795
car	99	200	2854
office	89	76	1023
street	57	78	1102
subway	49	22	265

Table 1. Amount of audio data for each context in NXP Software: number of files, duration available sub-clips

3.2. Protocols and metrics

For the experiments, the recordings were divided into shorter length sub-clips, with training and testing being performed on these. For the purpose of the classification system, these sub-clips represent individual training and testing samples. The segment length was selected to be 4 seconds. This segmentation allows implementation of an on-line classification system, offering a decision on the current environment with a 4-second latency.

The evaluation criteria is the global accuracy, averaged on five-folds partition, in conjunction with the memory size dedicated to SVs. We present the evolution of the accuracy and the memory size with different dataset reduction ratios. We check the statistical significance of the differences between methods using *Mc Nemar's test*. We compare the two different strategies used for reducing the size of the training set, and implicity the memory size and the complexity: one which uses k-means clustering for selecting the subset for training, and another which selects the subset randomly.

3.3. Implementation details

For the SVM we use the well known *LibSVM* library [18], using *rbf* kernel and a grid search to find the best C and γ parameters. The MFCCs are extracted for each audio sample using 40 bands, in 32 ms frames with a 50% overlap. We extract 13 MFCCs per frame, and then calculate the mean and variance over the length of the audio sample, which is 4 seconds. Each audio sample is represented by a 26-dimensional feature vector. During our experiments, the reduction from 26 to 13 features with LDA has shown to be a good choice, reducing the feature vector dimension to half without reducing the classification performance.

4. RESULTS

We consider the classification accuracy of the system using all training data available and compare it with the accuracy achieved when the training set is reduced to different ratios. For a given reduction rate, we evaluate the classification accuracy, the number of support vectors and memory requirements for the model.

train set size (% reduction)	SVs	accuracy	memory(KBytes)
480(0%)	276	0.51	11
475(10%)	275	0.51	11
263(50%)	173	0.51	6
192(70%)	137	0.49	5
103(90%)	88	0.50	3

Table 2. Accuracy vs number of support vectors and memory requirements obtained for different amounts of reduction. DCASE dataset.

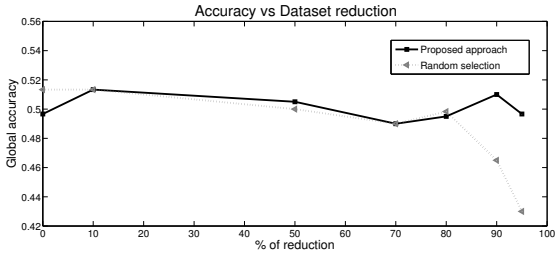

Fig. 2. Proposed method vs random selection. DCASE dataset.

4.1. DCASE dataset results

The proposed method was primarily designed for a large dataset case, where a big amount of SVs does not allow an *on-device* implementation and the complexity is a constraint. However, we have decided to test it also on DCASE dataset, even given its small size and no need for SVs reduction, for two reasons: to refer to some public available dataset, with whom other approaches can be compared, and also to test the proposed method with a limit case, where the dataset is too small. Results on DCASE dataset are presented in Table 2.

The two different training point selection methods are illustrated in Fig. 2. For this dataset, only after 90% of reduction, our method outperforms the random selection. At this point, the statistical significance is confirmed by McNemar's test. When drastically reducing the training set, the fact that we are using the cluster centroids as training points has an important effect on the modeling of classes, because the centroids represent with few samples the entire distribution of the class.

4.2. NXP Software dataset results

Results obtained for the NXP Software dataset are presented in Table 3 using the proposed method. The comparison with the system not using clustering is presented in Fig. 3. It can be observed that the proposed method for reduction outperforms the random selection from about 50% reduction level. We test the hypothesis that the two systems are equivalent in performance using McNemar's test: at a level of 70% reduction, the hypothesis can be rejected, so the difference in accuracy between the two is statistically significant.

train set size (% reduction)	SVs	accuracy	memory(KBytes)
5875(0%)	1396	0.73	72
5305(10%)	1282	0.73	66
2946(50%)	790	0.72	41
1782(70%)	524	0.71	27
604(90%)	225	0.68	11

Table 3. Accuracy vs number of support vectors and memory requirements for different amounts of reduction. NXP Software training set.

SVs(% reduction)	bus	car	office	subway	street
1396(0%)	0.75	0.79	0.91	0.54	0.64
1282(10%)	0.74	0.79	0.91	0.53	0.65
790(50%)	0.74	0.80	0.92	0.52	0.61
524(70%)	0.73	0.74	0.87	0.52	0.64
225(90%)	0.69	0.79	0.86	0.45	0.53

Table 4. Accuracy vs number of support vectors context-wise. NXP Software dataset

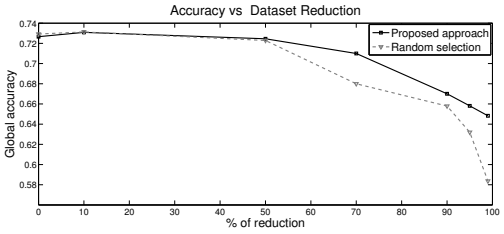

Fig. 3. Proposed method vs random selection. NXP Software dataset.

The first consideration concerns the reduction of SVs from the original system to 90%: a 5% loss in overall accuracy corresponds to 6 times smaller amount of SVs and almost 7 times lower memory requirements. The results show that it is possible to have a 1% loss in accuracy for half of the SVs and the memory. This ensures that the effect on the selection is minimum and part of the information retained in the training set is redundant.

The second consideration investigates context-wise evolution, presented in Table 4: all contexts have decreasing accuracy with the rate of reduction, except *car*. This is mainly because it the class with more training samples and the reduction doesn't affect this class so much. A similar behavior can be seen in *office*, but for a different reason: this environment has inherently less variation from a features point of view than other contexts. Finally, *subway* and *street* have a drop of almost 10%, due to less data samples and higher diversity within the same context.

5. CONCLUSIONS

In this paper we analyzed the effect of reducing the size of the training data for an SVM classifier with the aim to reduce the Support Vectors stored in the SVM model. Our initial problem was to reduce the SVM complexity for Audio Context Recognition on mobile devices. In our solution, we show that the Linear Discriminant Analysis is able to reduce the dimension of the features vectors without reducing the classification accuracy. Furthermore, k-means clustering allows control over the data points selection such that the selected training material covers the entire feature space. We have tested the proposed method on NXP Software and DCASE datasets, by evaluating the accuracy loss with respect to the reduction in the amount of Support Vectors. Our solution outperforms the random selection of training points and is scalable to different datasets. As further improvements, a smarter way to automatically select the initial number k of clusters could be investigated. Furthermore, the reduction could be done at different rates for different classes, to obtain a balanced representation of all classes in the model. We conclude that an important reduction of the number of support vectors can be achieved without a significant drop in classification accuracy. This translates into saving computational complexity and making possible an *always-listening* context awareness mode for mobile devices.

REFERENCES

[1] A.J. Eronen, V.T. Peltonen, J.T. Tuomi, A.P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi, "Audio-based context recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 1, pp. 321–329, Jan 2006.

[2] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. Laerhoven, and W. Van de Velde, "Advanced interaction in context," in *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*. 1999, HUC '99, pp. 89–101, Springer-Verlag.

[3] O. Räsänen, J. Leppänen, U. K. Laine, and J. P. Saarinen, "Comparison of classifiers in audio and acceleration based context classification in mobile phones," in *EUSIPCO-2011*, Sept 2011.

[4] G. T. Abreha, "An environmental audio-based context recognition system using smartphones," M.S. thesis, University of Twente, August 2014.

[5] D. Waltenegus, "Adaptive audio-based context recognition," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 39, no. 4, pp. 715–725, July 2009.

[6] M.W. Mak and S.K. Kung, "Low-power svm classifiers for sound event classification on mobile devices," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, March 2012, pp. 1985–1988.

[7] L. Bottou and C.J. Lin, "Support vector machine solvers," *Large scale kernel machines*, pp. 301–320, 2007.

[8] R. Koggalage and S. Halgamuge, "Reducing the number of training samples for fast support vector machine classification," *Neural Information Processing-Letters and Reviews*, vol. 2, no. 3, pp. 57–65, 2004.

[9] D. H. Mai and N. L. Chi, "Training data selection for support vector machines model," *IPCSIT vol.6*, 2011.

[10] Y.-J. Lee and S.-Y. Huang, "Reduced support vector machines: A statistical theory," *Neural Networks, IEEE Transactions on*, vol. 18, no. 1, pp. 1–13, Jan 2007.

[11] G. Roma, W. Nogueira, and P. Herrera, "Recurrence quantification analysis features for auditory scene classification," Tech. Rep., IEEE AASP Challenge: Detection and Classification of Acoustic Scenes and Events, 2013.

[12] B. Elizade, H. Lei, G. Friedland, and N. Peters, "An i-vector based approach for audio scene detection," Tech. Rep., IEEE AASP Challenge: Detection and Classification of Acoustic Scenes and Events, 2013.

[13] M. Chum, A. Habshush, and C. Rahman, A. Sang, "Scene classification challenge using hidden markov models and frame based classification," Tech. Rep., IEEE AASP Challenge: Detection and Classification of Acoustic Scenes and Events, 2013.

[14] L. Lu, S.Z. Li, and H.-J. Zhang, "Content-based audio segmentation using support vector machines," in *Multimedia and Expo, 2001. ICME 2001. IEEE International Conference on*, Aug 2001, pp. 749–752.

[15] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[16] T. Li, S. Zhu, and M. Ogihara, "Using discriminant analysis for multi-class classification: an experimental investigation," in *Knowledge and Information System*, Springer, Ed., 2006, vol. 10, pp. 453–472.

[17] D. Giannoulis, E. Benetos, D. Stowell, M. Rossignol, M. Lagrange, and M.D. Plumbley, "Detection and classification of acoustic scenes and events: An ieee aasp challenge," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*, Oct 2013, pp. 1–4.

[18] C.C. Chang and C.J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 1–27, 2011.