ACOUSTIC CONTEXT RECOGNITION FOR MOBILE DEVICES USING A 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

[‡] EURECOM
Biot, France

ABSTRACT

Automatic context recognition enables mobile devices to react to changes in the environment and different situations. While many different sensors can be used for context recognition, the use of acoustic cues is among the most popular and successful. Current approaches to acoustic context recognition (ACR) are too costly in terms of computation and memory requirements to support an always-listening mode. This paper describes our work to develop a reduced complexity, efficient approach to ACR involving support vector machine classifiers. The principal hypothesis is that a significant fraction of training data contains information redundant to classification. Through clustering, training data can thus be selectively decimated in order to reduce the number of support vectors needed to represent discriminative hyperplanes. This represents a significant saving in terms of computational and memory efficiency, with only modest degradations in classification accuracy.

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

1. INTRODUCTION

Context recognition aims to categorize the environment in which a computer system is used. The problem is particularly pertinent in the case of mobile devices given their use in multiple situations throughout the course of a typical day. Here, for instance, the ringer volume of a smart telephone might be adjusted according to whether the user is travelling on a bus, in an office or at home. The motivation stems from the continuous demand for advanced functionality by automatically adapting the device configuration to the situation.

Mobile devices are increasingly equipped with multiple, heterogeneous sensors, many of which provide cues useful to context recognition. Examples include light sensors, gyroscopes and accelerometers. Acoustic sensors are the most widely used in practice; almost every mobile device is equipped with a microphone. There is evidence that the use of acoustic cues outperforms context recognition with accelerometer measurements [1, 2] and that, in any case, complementary audio cues are useful in a fusion framework.

This paper is concerned with acoustic context recognition (ACR). Here, *context* refers to an ensemble of sounds, events and background noise. Many approaches are reported in the literature. While the majority use cepstral features typical of speech processing systems, a variety of different classifiers have been investigated. There exist distance based classifiers, such as k-nearest neighbor (kNN) [3], to classify examples based on distance to known data. In [4], Gaussian Mixture Model (GMM) classification is employed. One popular approach involves temporal modelling using hidden Markov models (HMMs) to classify the context through a sequence of events or states [5]. One of the last approaches represents a context as an histogram of audio events which are detected in a scene [6]. Others, more recent works [7, 8] shows that support vector machine (SVM) classifiers offer a better trade-off between high performance and low-complexity.

Efficiency is especially important with ACR for mobile devices. First, unreliable data connections and the power implications of continually communicating audio data to a remote server make cloud solutions impractical. While running locally on the device itself, computational efficiency is essential to minimise battery consumption. Second, the context is dynamic. The need for *always-active* ACR calls for algorithmic efficiency. Third, reliable context recognition usually requires context modelling with large amounts of data. Efficient modelling is thus needed to avoid the processing and storing in memory of large, complex models. None of the existing approaches to ACR meets all of these requirements. As an example, the complexity of an SVM classifier depends fundamentally on the number of training samples and the amount of support vectors (SVs) in the model. With large quantities of data being needed for reliable ACR, standard SVM classifiers are typically too complex.

This paper reports our work to develop an efficient ACR system for mobile devices. The general research hypotheses are that (i) gains in modelling efficiency can be achieved by reducing the redundancy in a large training dataset and that (ii) with only modest degradation in recognition performance, computational complexity can be reduced by using less complex models. Inspired by related research [9–11], the strategy

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

is to reduce the dependence of a typical ACR algorithm on a large training dataset. The main idea is to reduce computational complexity by purging a training dataset of samples deemed least relevant to the learning of decision boundaries.

The novel contribution in this paper relates to an additional level of data selection through clustering and decimation. The principal idea is to reduce model complexity by learning from a subset of training data selected uniformly from each cluster. The decimation of training data brings a sympathetic reduction in the number of support vectors, less complex models and, in turn, further reductions in computational requirements.

The rest of the paper is organized as follows: Section 2 describes the methods used for model complexity reduction; Section 3 presents the experimental set-up, database description and implementation details results. The results are analyzed in Section 4; Section 5 presents conclusions and directions for further investigation

2. REDUCED COMPLEXITY ACR

Complexity reduction is achieved through 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 classification. The steps of the system are presented in Fig. 1.

2.1. Feature extraction and selection

In the first step, full audio samples are first divided into fixed-length non-overlapping segments. This is common practice [12] as a means of improving granularity. Each segment is then divided into a sequence of short, over-lapping segments before Mel-frequency Cepstral Coefficients (MFCCs) are extracted. While designed for speech signals, MFCCs are popular for ACR and encode the spectrum of a signal into a compact and uncorrelated representation. The mean and standard deviation of the set of MFCCs for each segment is then determined such that each of them is then represented by a single, fixed-length feature vector.

Linear discriminant analysis (LDA) is applied in order to reduce feature vector size while improving discrimination. LDA is a supervised feature transformation technique which utilises class label information to identify a linear projection. Original features are projected into a new sub-space where the ratio of *between-class* variability to *within-class* variability is maximized according to the following cost function:

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

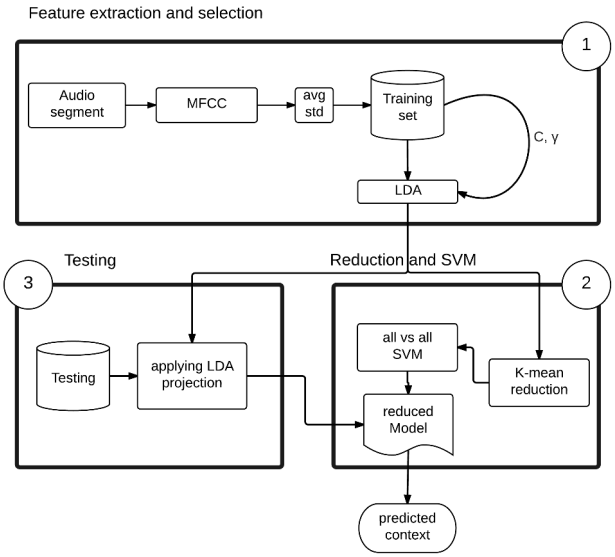

Fig. 1. The entire process of complexity reduction: **1.** feature extraction and selection using LDA. **2.** The SVM training, after the K-means dataset reduction **3.** The testing with SVM reduced model.

where S_b and S_w are the *between-class* and *within-class* scatter matrices calculated in the usual way [13]. Equation 1 is treated as a regular eigenvalue problem, where the eigenvectors corresponding to the largest eigenvalues are used to determine discriminant feature transformations [14]. LDA projections are learned using an independent training subset and applied without modification to test samples before classification. While LDA may not necessarily improve classification accuracy, dimensionality reduction reduces the size of resulting class models, therefore saving memory.

2.2. Training set reduction and SVM learning

The second step involves the learning of class models. First, the training data is clustered in order to select a sub-set of training samples for modelling.

The training data for a given class or context consists of n samples $x_i, i = 1...n$. The data is clustered into k clusters using a standard K-means algorithm which minimizes the average distance between the set of samples and a set of clusters centres $\mu_i, i = 1...k$ expressed as an objective function:

$$\arg\min_{C} \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_i||^2 \quad (2)$$

where $x \in C_i$ is the set of samples belonging to cluster i and μ_i is the i^{th} cluster mean. The cluster centroids are initialized randomly. The K-means algorithm is iterative, attributing samples at each iteration to its nearest cluster. Cluster

centres are updated and the algorithm is repeated until convergence. The data attributed to each cluster is then decimated according to random selection so that the full data distribution is now represented by a subset of the original data.

Clustering and data selection is performed for each class or context before a multi-class SVM classifier is trained with the reduced subset. The decimated training set reduces the number of support vectors (SVs) required to represent the SVM discrimant hyperplanes. This effectively reduces the size of the context models required for ACR.

2.3. Testing

The final step involves testing. Feature extraction is applied in the same way as before to each test audio sample. The same LDA projection is applied to reduce the feature vector dimension and to project each test sample into the same feature sub-space. Finally, test samples are classified according to the reduced set of support vectors for each context model.

3. EXPERIMENTS

The proposed method was assessed using two different databases, using five-fold partitioning into independent training and testing sets. Results are averaged across the five folds and each context. The baseline is the classification performance without reduction of the training set. In order to demonstrate the benefit of clustering, results are also presented for a similar system which reduces the training data set by random data selection without clustering.

3.1. Databases

The DCASE challenge dataset [15] consisting of 100 audio recordings, each of length 30 seconds. There are 10 different acoustic contexts and 10 recordings for each. Through other experiments not reported here, the DCASE dataset was found to be too small to explore fully the merit of the proposed approach. While the size of the DCASE dataset does not necessitate data decimation, it is included here since it is a standard database and thus supports the comparison of results generated by other researchers. Accordingly, results are also reported for a more extensive, though non-standard database collected by NXP Software. The NXP Software database was recorded by volunteers using mobile devices on which a recording application was installed. The application handles both data collection and labelling before uploading both to a centralised server. The recorded data covers five of the most common, everyday acoustic contexts: inside a bus, inside a car, office, subway and street. The amount of data available for each context is presented in Table 1.

3.2. Protocols and metrics

Each of the samples in both DCASE and NXP Software databases were divided into segments as described in Section 2 and subsequently treated as individual samples. While they are not independent, together they represent greater variability; this information would otherwise be lost in the averaging process applied during feature extraction. The division in to short segments also allows assessments reflective of on-line classification system, here offering a decision on the current context.

Context	Files	Duration (minutes)	Segments
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 the NXP Software database: number of files, recording duration and number of segments.

The evaluation criteria is the global recognition accuracy, averaged on five-fold partitions. The memory required to store SVs is a second evaluation criteria. Recognition accuracy and memory size are expected to be inversely related. The statistical significance between different recognition accuracies is determined according to a *McNemar* test [16].

3.3. Implementation details

SVM classifiers were implemented with the well known *LibSVM* library [17], using RBF kernels and a grid search to otimise C and γ parameters. Features are extracted from frames of 32 ms in duration with a 50% overlap. They comprise the mean and variance of 13 MFCCs extracted from each frame over the entire sample. The 26-dimensional feature vector is then reduced to 13th order through LDA projection with negligible impact on classification performance.

4. RESULTS

Classification accuracy is assessed using different rates of data decimation. In all cases, classification accuracy is assessed together with the number of SVs. As demonstrated in [18], memory and computational time of non-linear kernel SVM are $O(nd)$, where n denotes the number of SVs and d the features dimension. The size of memory has been calculated supposing 4 Bytes for each dimension of each SV.

4.1. DCASE dataset

Results for the DCASE dataset are presented in Table 2. They show that, even when the amount of training data is reduced by 90%, there is a negligible degradation in recognition accuracy, while the memory requirements are reduced by over 70%. Results for the two different decimation approaches are illustrated in Fig. 2. They show that the proposed approach

train set size (% reduction)	SVs	accuracy	memory (KBytes)
480(0%)	276	0.51	14
475(10%)	275	0.51	14
263(50%)	173	0.51	9
192(70%)	137	0.49	7
103(90%)	88	0.50	5

Table 2. Recognition accuracy, number of support vectors and memory requirements for different amounts of training data reduction, for the DCASE dataset.

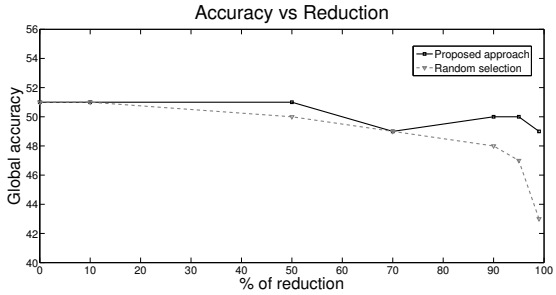

Fig. 2. An illustration of the different recognition accuracy for the proposed approach and random data selection. Results illustrated for the DCASE dataset.

significantly outperforms random data selection when the degree of reduction exceeds 80%. The McNemar test has been evaluated to reject the hypothesis that the results from different decimations are equal. The test confirms the statistical significance of these results, when the reduction is more than 80%. A complete benchmark of DCASE challenge has been presented in [19], where the different algorithms are compared. The current DCASE baseline (MFCC and GMM as classificator) has 55% of accuracy, while the best methods [20] has reached 71%. The drop of our system compared to them is due to different protocol (we are not considering the 30 seconds, but smaller sub-clips) and to more complex features. With same initial conditions, our system has 60% of accuracy.

4.2. NXP Software dataset

Results for the NXP Software dataset are presented in Table 3. While the degradation is more significant than for the DCASE dataset, a 90% reduction in training data and 85% reduction in memory requirements sill only causes a 5% drop in recognition accuracy from 73% to 68%. Results also show that if the number of SVs and required memory halved, the difference in performance is only 1% absolute. These results confirm the redundancy in the training data which, when removed, causes only negligible degradations in recognition accuracy while greatly reducing memory requirements.

Results for the two different decimation approaches are

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. As for Table 2, except for the NXP Software dataset.

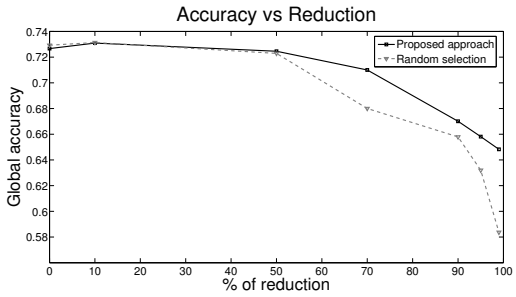

Fig. 3. Proposed method vs random selection. NXP Software dataset.

illustrated in Fig. 3. In this case the proposed approach significantly outperforms the random selection approach when the training data is reduced by more than 50%. The significance of these results is confirmed with a McNemar test.

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. NXP Software dataset results with context-wise accuracy

Finally, Table 4 illustrates the variation in performance for different contexts. With the exception of the car context, performance degrades as the training data is reduced. Results for subway and street contexts show the most significant degradations. This behaviour is likely caused by the different variation present in each context.

5. CONCLUSIONS

This paper presents a new, reduced complexity approach to acoustic context recognition for mobile devices. The principal idea involves the selective decimation of training data such that a reduced set of support vectors are required for classification, which then involves less memory and less computation. Linear discriminant analysis is applied to reduce

the dimension of the feature space without degrading classification accuracy. K-means clustering is the basis for data selection, ensuring that the full feature space is adequately represented after decimation. Evaluation on one small, standard dataset and one larger, non-standard dataset confirm that the decimation has only a modest impact on classification accuracy. Furthermore, contrastive experiments with a random selection approach demonstrate the merit of clustering prior to decimation. Since the degradation in results seems to depend on the context, further work should investigate context-dependent cluster optimisation, including the number of clusters and cluster initialisation. A complementary strategy involves decimation optimised at the class and cluster levels. The work shows that a significant reduction in memory and computational requirements can be delivered without significant impacts on classification accuracy. This approach paves the way for *always-active* context awareness for mobile devices.

REFERENCES

[1] 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.

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

[3] G. T. Abreha, "An environmental audio-based context recognition system using smartphones," Master's thesis, University of Twente, August 2014.

[4] H. Lu, W. Pan, N. Lane, T. Choudhury, and A. T. Campbell, "Soundsense: Scalable sound sensing for people-centric application on mobile phones," *MobiSys'09*, 2009.

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

[6] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Audio context recognition using audio event histograms," in *In Proc. European Signal Processing Conference*, 2010.

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

[8] M. Perttunen, M. Van Kleek, O. Lassila, and J. Riekki, "Auditory context recognition using SVMs," in *The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, UBICOMM*, Sept 2008, pp. 102–108.

[9] 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.

[10] D. H. Mai and N. L. Chi, "Training data selection for Support Vector Machines model," *IPCSIT vol.6*, 2011.

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

[12] L. Lu, S. Li, and H.-J. Zhang, "Content-based audio segmentation using Support Vector Machines," in *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001*, Aug 2001, pp. 749–752.

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

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

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

[16] J. L. Fleiss, B. Levin, and M. C. Paik, *Statistical methods for rates and proportions; 3rd ed.*, ser. Wiley Series in Probability and Statistics. Hoboken, NJ: Wiley, 2003.

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

[18] H. Cao, T. Naito, and Y. Ninomiya, "Approximate RBF Kernel SVM and Its Applications in Pedestrian Classification," in *The 1st International Workshop on Machine Learning for Vision-based Motion Analysis - MLVMA'08*, Marseille, France, Oct. 2008.

[19] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification," *CoRR*, vol. abs/1411.3715, 2014.

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