TUT ACOUSTIC SCENE CLASSIFICATION SUBMISSION

Gonçalo Marques

Instituto Superior de Engenharia de Lisboa
Electronic Telecom. and Comp. Dept.
R. Conselheiro Emídio Navarro 1
1959-007 Lisboa, Portugal
gmarques@deetc.isel.pt

Thibault Langlois

Faculdade de Ciências da Universidade de Lisboa
Informatics Dept., Edifício C6, Piso 3
1749-016 Lisboa, Portugal
tl@di.fc.up.pt

ABSTRACT

This technical report presents the details of our submission to the D-CASE classification challenge, Task 1: Acoustic Scene Classification. The method used consists in a feature extraction phase followed by two dimensionality reduction steps (PCA and LDA) the classification being done using the k nearest-neighbours algorithm.

Index Terms— Machine Learning, Signal Processing, Music Information Retrieval, Bag of Frames.

1. INTRODUCTION

The system we purpose for the TUT Acoustic Scene Classification challenge is a classical classification system in the sense that it uses typical machine-learning data transformation and classification algorithms in the decision making process. First, each audio excerpt is converted into a single feature vector which is the representation of choice for standard machine-learning methods. Then, the whole dataset is transformed via principal component analysis (PCA), an unsupervised dimensionality reduction technique, followed by a linear discriminant analysis (LDA) projection. LDA is a supervised process, and the projection tries to maximize the ratio between intra and inter class scatter, but it is not a classification method since no decision is involved. For classification, a k nearest-neighbours (k-NN) algorithm was used. The experimental configuration used in our tests is common in many audio classification works (or at least parts of it – see for *e.g.* [1, 2, 3, 4] among many others) and therefore it does not bring any original contribution in terms of the algorithmic setup. In fact, our system falls under the standard "bag of frames" classifiers commonly used in music information retrieval applications, and in that sense is just another baseline system that can complement the results in [5]. We used the same data partitioning and cross-validation setup provided with the database and our results, 78.2%, are 5% above the ones in [5] (see Section 4). However, the experiments we conducted also revealed some unexpected variations in terms of accuracy when the whole dataset or just part of it was used to estimate the PCA and LDA projections. This is an indication that there may be differences between feature class-dependent distributions among folds. The structure of the remainder of this report is as follows: Section 2 describes the data and the feature extraction process used in our experiments, Section 3 describes our approach to acoustic scene classification, followed by Section 4 where we present our results.

2. DATA AND FEATURE EXTRACTION

The dataset used in this work was create in the context of the DCASE2016 challenge [6] for the acoustic scene classification task. The dataset contains 1170, 30-seconds audio excerpts from the following acoustic scenes: Beach, Bus, Café/Restaurant , Car, City Center, Forest Path, Grocery Store, Home, Library, Metro Station, Office, Urban Park, Residential Area, Train, and Tram. The dataset divided into four folds for cross-validation testing. We used the same data partition in our experiments and our results are averaged over the four test folds.

The features used are the all-purpose Mel frequency cepstral coefficients (MFCCs) representation. The audio was divided into 23 ms segments (1024 samples at 44.1 kHz) with 50% overlap, and we used 100 Mel bands to extract 23 MFCCs plus the zero order MFCC and the frame's log-energy, plus the delta and acceleration coefficients. This means that the audio is converted into a sequence of $25 \times 3 = 75$ dimensional vectors. We used the software VoiceBox [7] to extract the features. In order to convert each audio excerpt into a single feature vector, the sequence of MFCC features is summarized using the median and logarithmic standard deviation. The median was used instead of the mean since this statistic is more robust to outliers. The log-standard deviation is given by $20 \log_{10}(\sigma_i)$ where σ_i is the standard deviation of feature i (with $i = 1,\ldots,75$). The reason to use the log-standard deviation instead of plain standard deviation was to convert these feature values to an order of magnitude comparable the median feature values - otherwise the standard deviation values would be a few orders of magnitude lower, and during the PCA pre-processing step, this dimensions would be discarded as noise since they would not contribute in any significant way to the overall data variance. The statistics return two 75-dimensional vectors which are concatenated, so each audio excerpt is represented by a 150-dimensional feature vector.

3. METHOD

The proposed classification approach is divided into three main blocks: feature pre-processing via principal component analysis, feature transformation by linear discriminant analysis and finally a classification step performed by a k-nearest neighbour classifier.

Principal Component Analysis: PCA is a standard dimensionality reduction technique, where the data is decorrelated by projecting it into orthogonal directions of maximum variance. These directions, the principal components, are obtained using a eigen-decomposition of the data covariance matrix, and in our experi-

ments we kept enough components to explain 99.9% of the total data variance. The PCA-transformed data was also whitened - each data dimension was scaled in order to have unit variance.

PCA is an unsupervised learning method, and therefore it is common to used the whole dataset to estimate the principal components. Nevertheless, in many applications it is not practical to recalculate the covariance matrix every time a new signal is recorded. We performed some tests in order to have an idea of how the classification performance is affected by using just part or the whole dataset. The results (see Table 4) show that there is no significant decrease in accuracy (less than 1%) when the PCA projection is estimated with only the training set.

Linear Discriminant Analysis: LDA is commonly used as a preprocessing step for pattern classification. It is also a dimensionality reduction technique since the data is projected into $c-1$ dimensional space where c is the total number of classes ($c=15$ for this challenge).

LDA is a supervised learning method, and therefore the projection should be calculated with the training set only, otherwise we are indirectly including information about the class labels in the test set. Estimating the LDA projection with the whole dataset can result in overly optimistic performance values. More so if there is a relatively high number of classes and a relatively low number of examples, as in the case of this challenge dataset. We tested the performance of our system using the whole dataset to estimate the LDA projection in order to gauge the increase in performance compared to the "correct" evaluation procedure. The results showed a significant increase in performance, which in our perspective, is somewhat surprising. These are presented in Section 4.2, along with a discussion on possible causes of such a performance discrepancy.

k-Nearest Neighbours: k-NN is an instance-based learning, where class membership is assigned based on a majority vote of its neighbours. k-NN is possibly one of the simplest classification methods, and therefore it is well suited for a baseline system. We tested two distance metrics with the k-NN algorithm, the cosine and the Euclidean distance, and opted for the Euclidean distance because it yielded slightly better accuracy results. We also ran the algorithm with different number of neighbours (from 5 to 31 - using a increment of two) and chose empirically $k=9$. The results reported in Section 4 are obtained using the Euclidean distance metric, and $k=9$.

4. EXPERIMENTAL RESULTS

This section is divided into two parts. In the first, we present the results obtained with our method. The experimental setup is described, the system performance is measured in terms of accuracy, either with mean or class specific values. In the second, we present the performances obtained when the whole dataset is used to estimate the LDA projection. This is not the correct procedure to estimate our system performance. The intent is to have an idea of by how much the performance values are inflated.

4.1. System Performance

The results presented in this section were obtained using the following experimental setup. The PCA projection was calculated using the whole (development) dataset, while the LDA projection was estimated using only the training set. In our tests, we used 4-fold cross validation and the same data partitioning provided with the dataset. This means that a total of four LDA projections where estimated with three training folds. The presented result pertain to the tests folds only. The average accuracy obtained was 78.2%. In Table 2 are the (average) accuracies per class. These context-wise performances vary from 52.6% (Train class) to 93.6% (City Center and Metro Station classes). Table 3 shows the accuracies per fold and Table 4 are the same results for the case where the PCA projection is obtained using only the training set data.

60	0	0	0	**1**	**2**	0	0	**1**	0	**1**	**9**	**3**	0	**1**
0	**52**	**6**	**2**	0	0	**1**	0	**5**	0	0	0	0	**11**	**1**
0	0	**62**	0	0	**2**	**8**	0	**2**	**4**	0	0	0	0	0
0	**3**	0	**66**	0	0	0	0	**1**	0	0	0	0	**7**	**1**
1	0	0	0	**73**	0	0	0	**1**	**1**	0	0	**2**	0	0
1	0	0	0	0	**68**	0	0	0	0	**2**	**7**	0	0	0
0	0	**4**	0	0	0	**64**	0	**1**	**9**	0	0	0	0	0
2	**2**	**7**	0	0	**2**	0	**50**	**8**	0	**6**	0	0	0	**1**
0	0	**9**	0	0	0	**1**	0	**68**	0	0	0	0	0	0
0	0	0	0	0	0	**2**	0	**3**	**73**	0	0	0	0	0
0	0	0	0	0	**1**	0	**5**	0	0	**72**	0	0	0	0
2	**1**	0	0	**1**	**13**	**3**	0	**6**	0	0	**47**	**5**	0	0
2	0	**2**	0	**1**	**3**	0	0	0	0	**1**	**17**	**51**	0	**1**
0	**11**	**8**	0	0	**2**	**4**	0	0	**5**	**1**	0	0	**41**	**6**
0	**1**	0	0	0	0	**7**	0	0	0	**1**	0	0	**1**	**68**

Table 1: Confusion matrix - the rows are the true classes, the columns are the classification results. The class order is the same as the one given in Table 2: in the first row are the samples from the class Beach, in the second from class Bus, and so on. This matrix was obtained by the sum of the four confusion matrices - one per test fold. A perfect classification would result in this matrix having the value 78 on the main diagonal (number of examples per class) and zeros for the rest of the coefficients.

1. Beach	76.9%
2. Bus	66.7%
3. Café/Restaurant	79.5%
4. Car	84.6%
5. City Center	93.6%
6. Forest Path	87.2%
7. Grocery Store	82.1%
8. Home	64.1%
9. Library	87.2%
10. Metro Station	93.6%
11. Office	92.3%
12. Urban Park	60.3%
13. Residential Area	65.4%
14. Train	52.6%
15. Tram	87.2%

Table 2: Accuracy per class. Accuracy values obtained with the mean of all four test folds.

Table 1 shows the confusion matrix (obtained summing the four confusion matrices in each test fold). Each line refers to the examples of a single class; the class order is the same as the one in Table 2. In the columns are the classification results. For example, for the class Beach, 60 audio excerpts were correctly classified, 9 were classified as the class Urban Park, and nine others were also misclassified. The results also reveal some particular error corre-

lations among certain classes. For instance many Residential Area samples were misclassified as Urban Park (a total of 17 errors). Five Urban Park pieces were attributed to Residential Area, however the excerpts o this class have a higher tendency to get confused with another class: Forest Path (a total of 13 errors). These mislabelling seems understandable since these acoustic scenes share some resemblances. Another example of classes that have similar acoustic characteristics and a high number of errors between them are Bus and Train. Other relations that seem to make some sense could be found such as the case of Beach and Urban Park, or Home and Library, but further tests would be needed to determine if a real correlation exists.

	Fold 1	Fold 2	Fold 3	Fold 4
Accuracy	79.0%	72.1%	82.9%	78.8%

Table 3: Accuracy per fold. The mean accuracy is 78.2%.

	Fold 1	Fold 2	Fold 3	Fold 4
Accuracy	79.3%	71.7%	82.6%	76.0%

Table 4: Accuracy per fold. The mean accuracy is 77.4%. This results were obtained using only the training set to estimate the PCA projection.

4.2. LDA estimation with the whole dataset

In this section we discuss the results obtained when we used the whole dataset to estimate the LDA projection. This is not the correct testing methodology, since, when doing this, we are implicitly including test label information in the model training process. The intent here is just to determine by how much the performances are over evaluated when using this incorrect experimental procedure.

In Table 6 are the accuracies per test fold. The mean accuracy is 90.8% which is 12.6% points higher than our baseline system. This is also an indication that there is a some variability of class-dependent feature distributions among folds. The partition process used for this dataset [5] may be the cause, since it was based on recording location. This division was done in order to avoid overestimating systems performances, since in this way, segments from a single recording are assigned to only one fold. We believe that the variability between folds is also due to the relatively low number of examples per class, and increasing the number of examples in the database will reduce this variation.

Table 5 shows the confusion matrix. There is some similarities to the error patterns found in Section 4.1. For instance, the Residential Area samples are still misclassified as Urban Park, Urban Park as Forest Path, and Home as Library. Other errors though, like the confusion between Bus and Train classes, have almost vanished.

5. CONCLUSION

The submitted proposal performs better that the base line (78.2% vs 72.5%) on the development dataset. We look forward to see the performance of this simple approach on the evaluation dataset. In this regard, the classifications for the evaluation dataset were obtained after performing the PCA and LDA on the four development folds. On the downside, the k nearest-neighbors has the evident drawback of not being scalable. We believe that the performance

67	0	0	0	**1**	**2**	**1**	0	**1**	0	0	**2**	**3**	0	**1**
0	**72**	**3**	**1**	0	0	0	0	**1**	0	0	0	0	**1**	0
0	0	**77**	0	0	0	**1**	0	0	0	0	0	0	0	0
0	**1**	0	**77**	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	**76**	0	0	0	0	0	0	0	**1**	0	0
0	0	0	0	0	**75**	0	0	0	0	0	**3**	0	0	0
0	0	**1**	0	0	0	**73**	0	0	**4**	0	0	0	0	0
1	**1**	**2**	0	0	0	**1**	**61**	**8**	0	**3**	0	0	0	**1**
0	0	**2**	0	0	0	0	0	**76**	0	0	0	0	0	0
0	0	0	0	0	0	0	0	**1**	**77**	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	**78**	0	0	0	0
2	0	0	0	0	**7**	**2**	0	**4**	0	0	**62**	**1**	0	0
1	0	0	0	0	**2**	0	0	**2**	0	0	**14**	**58**	0	**1**
0	**1**	**3**	0	0	0	**3**	0	**1**	**2**	**1**	0	0	**60**	**7**
0	0	0	0	0	0	**3**	0	0	0	**1**	0	0	**1**	**73**

Table 5: Confusion matrix obtained by the sum of the four confusion matrices - one per test fold. The results were obtained using (inappropriately) the whole dataset to estimate the LDA linear projection.

	Fold 1	Fold 2	Fold 3	Fold 4
Accuracy	95.9%	85.9%	92.3%	89.0%

Table 6: Accuracy per fold. The mean accuracy is 90.8%. This results were obtained using (inappropriately) the whole dataset to estimate the LDA linear projection.

obtained with this simple non parametric approach highlights the benefits dimensionality reduction (PCA and LDA).

6. ACKNOWLEDGMENT

Gonçalo Marques thanks FI-Sonic[1] for supporting his research.

7. REFERENCES

[1] S. Dieleman and B. Schrauwen, "Multiscale approaches to music audio feature learning," in *14th International Society for Music Information Retrieval Conference (ISMIR-2013)*. Pontifícia Universidade Católica do Paraná, 2013, pp. 116–121.

[2] C.-H. Lee, J.-L. Shih, K.-M. Yu, and H.-S. Lin, "Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features," *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 670–682, 2009.

[3] J. Salamon and J. P. Bello, "Unsupervised feature learning for urban sound classification," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 171–175.

[4] G. Wu, J. Zhu, and H. Xu, "A hybrid visual feature extraction method for audio-visual speech recognition," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 1829–1832.

[5] A. Mesaros, T. Heittola, and T. Virtanen, "Tut database for acoustic scene classification and sound event detection," in *24rd European Signal Processing (EUSIPCO'16)*, Budapest, Hungary, 2016.

[1] http://www.fi-sonic.com/

[6] http://www.cs.tut.fi/sgn/arg/dcase2016/.

[7] VOICEBOX: Speech Processing Toolbox for MATLAB (2005) by Mike Brookes.