EUSIPCO 2013 1569742319

SEMI-SUPERVISED LEARNING FOR MUSICAL INSTRUMENT RECOGNITION

Aleksandr Diment, Toni Heittola, Tuomas Virtanen

Tampere University of Technology
Department of Signal Processing
Korkeakoulunkatu 1, 33720, Tampere, Finland

ABSTRACT

In this work, the semi-supervised learning (SSL) techniques are explored in the context of musical instrument recognition. The conventional supervised approaches normally rely on annotated data to train the classifier. This implies performing costly manual annotations of the training data. The SSL methods enable utilising the additional unannotated data, which is significantly easier to obtain, allowing the overall development cost maintained at the same level while notably improving the performance. The implemented classifier incorporates the Gaussian mixture model-based SSL scheme utilising the iterative EM-based algorithm, as well as the extensions facilitating a simpler convergence criteria. The evaluation is performed on a set of nine instruments while training on a dataset, in which the relative size of the labelled data is as little as 15%. It yields a noteworthy absolute performance gain of 16% compared to the performance of the initial supervised models.

Index Terms— Music information retrieval, musical instrument recognition, semi-supervised learning

1. INTRODUCTION

Musical instrument recognition belongs to the music information retrieval research area, and its applications include, e.g., automatic music database annotation for indexing and retrieval purposes and automatic music transcription applications, which could benefit from identifying the instruments present in the recording. Furthermore, one may think in broader terms and realise applicability of musical instrument recognition for other areas, such as musical genre classification, where instrumentation may serve as a feature [1]. The recent advancements in the area of instrument recognition include the use of multiscale mel frequency cepstral coefficients [2] and projective non-negative matrix factorisation [3].

Instrument identification is often treated as a supervised classification problem requiring annotated data in order to train a classifier, as opposed to unsupervised tasks, which operate on unannotated data. To obtain such annotated datasets is quite laborious: even though there are limitless possibilities to collect the audio, its annotation requires tedious and expensive human work. Therefore, a requirement of a technique that would overcome this complication is quite apparent.

Semi-supervised learning (SSL) addresses the requirement of large datasets needed to train a classifier to demonstrate a sufficient level of generalisation capability. Basically, the larger and more diverse the training dataset is, the better generalisation properties one may expect. In SSL, this dataset extension is approached by incorporating additional data that is not annotated. Semi-supervised techniques have shown to be a successful approach in numerous machine learning tasks, such as text classification, computer vision, network traffic classification, as well various audio-related problems, including gender and speaker identification [4], prosodic event detection [5] and sound event classification [6]. In the area of music information retrieval SSL has been also used, e.g., for music artist style identification [7], music genre classification [8] and note onset detection [9]. However, they have not yet been applied for musical instrument recognition problem. The related works within neighbouring areas deal with one of the SSL techniques for singing voice detection [10] and an idea of weak labelling (where a label indicates appearance or absence of an instrument in a mixture) for instrument recognition [11].

The primary objective of this work is to show whether SSL is capable of introducing improvement in the performance of an instrument recogniser. It starts with an description of the applied feature, as well as the details of the semi-supervised training and recognition (Section 2). The SSL concept is studied on an example of the iterative EM-based algorithm. Furthermore, its extensions that facilitate a smoother transition between the models along the iterations are applied. Section 3 presents the details of the evaluation of the method on an example of a nine instrument note-wise classification case, which is followed by the conclusions drawn in Section 4.

2. SEMI-SUPERVISED LEARNING FOR MUSICAL INSTRUMENT RECOGNITION

This section presents the details of the developed system, which applies SSL for instrument recognition. It starts with the description of the utilised features followed by the implementation details of the recogniser and the training algorithm.

2.1. Feature extraction

As features, the static and delta mel frequency cepstral coefficients (MFCCs) are utilised for representing the timbre of the musical instruments. Originally, MFCCs have been used in speech recognition. With their aid it is possible to extract the information about the spectral envelope in order to detect formants, which characterise the speech content.

In the musical instrument signals, however, the presence of formants in the spectrum is rarely strong [12], or they are not a factor independent from fundamental frequency, in contrast to speech signals. Still, the MFCCs have shown quite satisfactory performance in musical instrument classification [13] and proven to be amongst the most effective features applied in this area [14] due to their ability to parametrise the rough shape of the spectrum, which is different for each instrument. Furthermore, being a perceptually-based representation of an acoustic signal, MFCCs are applicable to characterise any perceptually meaningful sound, including speech, musical instruments and natural ambience [15].

2.2. Recogniser

The implemented system performs supervised as well as semi-supervised learning, and both scenarios utilise mixture models, namely Gaussian mixture models (GMM). For each class the feature vectors obtained from the labelled data are used to train the GMM, i.e., to estimate the parameters of the mixture model that best explains these feature vectors. The EM algorithm [16] is applied for this purpose.

In recognition, the models obtained during the training stage are class-wise fit into each frame of each test instance producing log-likelihoods. The latter are subsequently summed over the frames of the test instance. Thereupon, the label of the class whose model has produced the highest log-likelihood is assigned to that instance.

In the supervised case, the training ends at the point when the GMMs are obtained based on the labelled training data. The semi-supervised training scenario continues by incorporating the unlabelled data and learning the new GMMs with the aid of an EM-based algorithm, which is described in the upcoming section.

2.3. Semi-supervised training

The training is performed with the aid of the conventional supervised EM-algorithm, as well as its SSL extension. In order to incorporate SSL into the EM algorithm for GMM, the iterative EM-based algorithm is applied, as presented in [4], as well as its further extended version (see Algorithm 1). It operates on the complete training dataset S comprised of the labelled S^l (with the labelled samples of indices $i = 1, \ldots, L$) and unlabelled S^u (with the unlabelled instances of indices $i = L+1, \ldots, L+U$) subsets, where L and U are the numbers of labelled and unlabelled samples, respectively.

Input: labelled data S^l, unlabelled data S^u.
Set $t = 0,\ t^* = 0$.
[Initial M] Initialise $\hat{\boldsymbol{\theta}}^{(0)} = \arg\max_{\boldsymbol{\theta}} P(S^l|\boldsymbol{\theta})$.
repeat
  Weight labelled data (Eq. 2).
  for $j^\dagger = 1, \ldots, M$ **do**
    [E] Set $\hat{\mathbf{z}}^{(t+1)} = E[\mathbf{z}|S; \hat{\boldsymbol{\theta}}^{(t)}]$.
    for $i = L+1, \ldots, L+U$ **do**
      Set $j^* = \arg\max_j \hat{z}_{ij}^{(t+1)}$.
      if $j^\dagger = j^*$ **then**
        Set $\hat{z}_{ij,j=1,\ldots,M}^{\text{hard}(t+1)} = \begin{cases} 1 & \text{if } j = j^* \\ 0 & \text{otherwise} \end{cases}$.
      else
        Set $\hat{z}_{ij,j=1,\ldots,M}^{\text{hard}(t+1)} = \hat{z}_{ij,j=1,\ldots,M}^{\text{hard}(t)}$.
      end
    end
    [M] Set $\hat{\boldsymbol{\theta}}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} P(S, \hat{\mathbf{z}}^{\text{hard}(t+1)}|\boldsymbol{\theta})$.
    Set $t = t + 1$.
  end
  Set $t^* = t^* + 1$.
until *convergence.*
Output: $\hat{\boldsymbol{\theta}}^{(t)}$.

Algorithm 1. *The iterative EM-based algorithm for SSL with the proposed extensions highlighted.*

Firstly, the supervised EM-algorithm is applied in training based on the labelled instances only in order to obtain the initial model parameters estimate $\hat{\boldsymbol{\theta}}^{(0)}$. Thereupon, the iterative algorithm incorporates the unlabelled data in such a manner that the expected values of the hidden variable z_{ij} (Equation 1) are used to estimate labelling for the unlabelled examples at each step. The hidden variable z_{ij} is defined for all $j = 1, \ldots, M$ class indices (where M is the total number of classes) and for all training samples $\mathbf{x}_i$, $i = 1, \ldots, L, L+1, \ldots, L+U$ as

$$z_{ij} = \begin{cases} 1 & \text{if } y_i = j \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where y_i is the true labelling of the instances, known for the labelled data and estimated in the case of unlabelled data. Combined over the whole training dataset, the instances of this hidden variable constitute a matrix $\mathbf{z}$, and its estimated value at the iteration $t+1$ is referred to as $\hat{\mathbf{z}}^{(t+1)}$. The labelled data together with the unlabelled data with $\hat{\mathbf{z}}^{(t+1)}$ are used to re-estimate the model parameters $\hat{\boldsymbol{\theta}}^{(t+1)}$, and the estimation of $\hat{\mathbf{z}}$ and $\hat{\boldsymbol{\theta}}$ is repeated iteratively.

2.3.1. Labelled data weighting

It has been noted [4] that the EM-based algorithms for SSL improve the recognition accuracy in case the initial labelled data size is relatively low. However, the difference in the advantages of incorporating large and small amounts of unlabelled data has been reported relatively insignificant, which is explained by the fact that with increasing unlabelled data the parameter estimates depend very little on the labelled data and reliable class information [4]. Therefore, it has been suggested to de-weight the impact of the unlabelled data, which can be achieved by scaling the contribution from unlabelled data in terms of the computed log-likelihoods. As a result, the certainty of the models that are obtained from labelled data is emphasised and the algorithm is more likely to rely on the certainly labelled data.

One could apply a somewhat coarser but simplified way of de-weighting the contribution of the unlabelled data. Its essence is in replicating the labelled data several times, and along the subsequent iterations the replication factor is gradually decreased:

$$S^l = w(t) \diamondsuit S^l, \tag{2}$$

where $w(t)$ is a decreasing weighting function of iteration index t and $\diamondsuit$ stands for the replication operation. This approach, albeit not as flexible, does not introduce additional complexity to the algorithm, while still expectedly emphasising the significance of the a priori correct labelling.

2.3.2. Class-wise retraining

The basic version of the iterative EM-based algorithm contemplates retraining of all the models at each iteration. It appears apparent that at any retraining stage there exists a classification error. If some data point was previously classified correctly leading to a somewhat correct model of its actual class, the classification error introduced at a later point would mean degradation of the models of two classes: the actual origin of the data point and the misclassification result.

In order to reduce such coupled model degradation effect, we suggest to apply a so-called *class-wise retraining* approach that enforces the previously obtained models of one class not to change while training another class. The essence of the approach is that at each iteration the models of only one class are retrained, while the others remain based on the previous labels (see Algorithm 1, where this extension is highlighted in the statements that handle the variables j^* and $t^\star$). As an additional outcome of the method, a smoother transition between the models is expected, thus resulting in fewer local peaks in the accuracy curve along the iterations of the algorithm, which could additionally benefit the criterion of convergence.

3. EVALUATION

We evaluate the performance of the implemented algorithm and its extensions in a separate note-wise instrument classification scenario. The following parameters of the feature extraction and training algorithms are used in evaluation: 16 MFCCs are extracted in 20 ms frames with 50% overlap, and each class is represented by a GMM of 16 components. Based on the preliminary experiments with different instrument sets, a sequence of values $[8, 6, 4, 2, 1, 1, \ldots]$ is used along the iterations of the extended algorithm as a labelled data weighting parameter w.

3.1. Acoustic material

The evaluation is performed with a dataset consisting of separate monophonic note recordings. A set of nine instruments originating from the RWC Music Database [17] is used. Each of them is represented by three instances, which stand for different instrument manufacturers and musicians. The instruments used in evaluation along with the number of separate note recordings originating from each of them are the following: Pianoforte (792), Classic Guitar (702), Electric Guitar (702), Electric Bass (507), Trombone (278), Tuba (270), Bassoon (360), Clarinet (360) and Banjo (941). This choice of instruments is influenced by the requirement of a sufficiently high number of notes per instrument for their adequately consistent representation in the database.

The dataset, consisting of 4912 recordings in total, is divided into the three following groups: the labelled training, the unlabelled training and the testing subsets, where the labelled-to-unlabelled ratio of the dataset sizes is set to 15/85. The labelled and unlabelled datasets are always acquired from different instrument instances in order to better resemble the real-life application scenario. For the testing set, a separate instance is used for the similar reasons, and the approximate ratio between the training and test set sizes is 70/30. The notes are mostly recorded in chromatic order, which may make the truncated datasets biased towards lower notes. To eliminate this, the notes within each set are also randomised.

3.2. Decision on convergence

In [4], the experiments were conducted using the models obtained after a single iteration of the EM-based iterative algorithm. The iterations may go on, but there is a need of a rule that would facilitate a decision to terminate the algorithm. One possible solution is to check the total number of labels that change at each iteration. One can utilise the matrix of the hidden variables $\hat{z}_{ij}^{\text{hard}}$ provided by the algorithm (Equation 1) to check for a label change count (LCC):

$$\text{LCC}^{(t+1)} = \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \left| \hat{z}_{ij}^{\text{hard}(t+1)} - \hat{z}_{ij}^{\text{hard}(t)} \right|, \tag{3}$$

Table 1. Experimental results, where *initial* stands for the accuracy obtained at the initial, purely supervised iteration of the algorithm based on the 15% of data, whereas the *final* results are produced by 40 semi-supervised (macro)iterations.

Test case	Recognition accuracy, %	
Supervised, 100% of data	83.8	
SSL with iterative EM	**initial**	**final**
basic algorithm [4]	61.4	76.5
with class-wise retraining	61.4	74.3
with lab. data weighting	61.4	75.1
with both extensions	61.4	77.0

where $\mathrm{LCC}^{(t+1)}$ — label change count at iteration $t+1$. By normalizing this value by the first label change, one can obtain a label change rate (LCR):

$$\mathrm{LCR}^{(t+1)} = \frac{\mathrm{LCC}^{(t+1)}}{\mathrm{LCC}^{(2)}} \cdot 100\%, \tag{4}$$

Moreover, the possible local minima in the LCR curve that may occur at the earlier stages of the algorithm may trigger the termination earlier than the actual convergence. These may be minimised with the aid of moving average filter based on the M preceding values, where M equals the number of classes. As a result, low values of the smoothed LCR are expected when the obtained models are sufficiently certain, which suggests that the algorithm may be terminated.

3.3. Results

The evaluation is performed in five different scenarios. Firstly, a fully supervised case is considered, i.e. all the data that is used in each evaluation scenario, is in this case labelled. The obtained results are expected to indicate an upper limit for the possible performance of SSL since it can be viewed as SSL with all the labels estimated correctly. The following four evaluation scenarios incorporate the non-modified iterative EM-based algorithm, as well as the proposed extensions and their combination.

The combined results are presented in Table 1, and the performance of the SSL algorithms along the iterations given the identical datasets is compared in Figure 1, which additionally includes the curves of the LCR. The term *macroiteration* in the case of the class-wise retraining approach stands for a set of iterations performed to retrain the models of all classes once and is referred to as t^* in Algorithm 1.

The LCR values are presented without the smoothing operation in order to enable a more detailed comparison of the approaches in terms of resulting model certainty. For the purpose of a fair comparison, the total number of performed iterations or macroiterations in all cases is set to be the same, i.e.

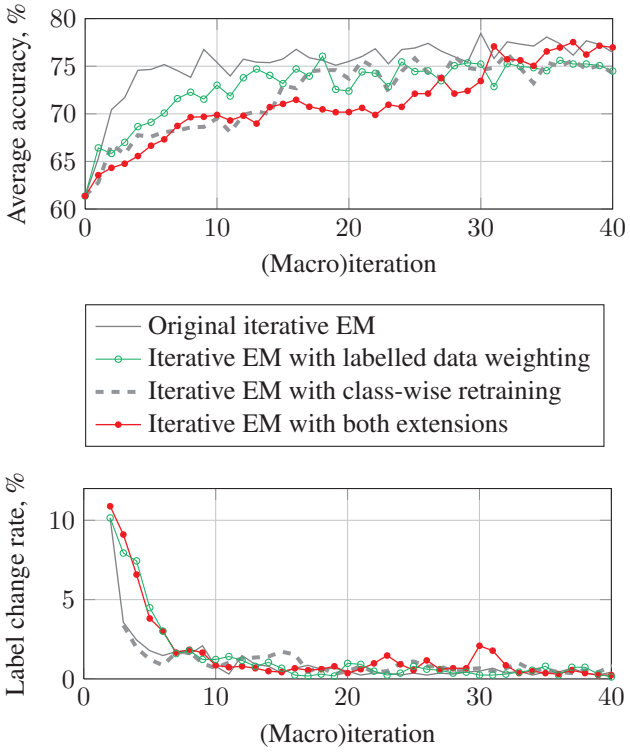

Fig. 1. Average accuracy and values of the LCR as functions of number of iterations or macroiterations.

the thresholding operation on the LCR has not been applied. However, all the curves demonstrate a similar behaviour of the LCR values and its variations being reduced along the iteration count axis in the region, where the accuracy curve reaches some degree of saturation. This suggests that, indeed, a value of the LCR being below a defined threshold could be a valid convergence criterion.

By examining the values of LCR with correspondence to the average accuracy values at the same iteration count, an observation can be drawn that the LCR curve reaches its stable low values at the point when the accuracy does not appear to have any potential to grow. Even the sudden peak in the LCR curve corresponding to the algorithm incorporating both extensions (macroiteration index 30) does not appear to be a shortcoming: it clearly indicates that the average accuracy will change somewhat significantly at the next iteration.

Generally speaking, all the evaluated algorithms are capable of yielding similar ultimate improvement (around 12-16%) of the initial supervised models at roughly the same time instance. The basic algorithm produces the most oscillating behaviour, although it reaches the convergence earlier. The proposed extensions, especially when incorporating class-wise retraining, produce a smoother transition between the models along the iterations and ultimately reaches the same performance level.

4. CONCLUSIONS

In this work, the applicability of SSL to the problem of musical instrument recognition has been explored. The basic EM-based SSL algorithm and its two proposed extensions facilitating a smoother transition between the learnt models and their increased certainty have been implemented and evaluated.

The evaluation performed on a nine-instrument note-wise classification case given as little as 15% annotated training data has shown up to 12-16% absolute improvement of the initial models' performance, which corresponds to the relative decrease of the error rate by 40%.

As a suggestion for the future investigation, a more sophisticated feature extraction method could be incorporated into the developed system. In the case of a more effective feature extraction, the impact of the semi-supervised techniques could be more apparent.

Moreover, a further investigation of the system's performance could be conducted in more complex scenarios by increasing the number of instruments in the set or by introducing noise, reverberation and distortions to the datasets in order to mimic the real-world application scenario.

5. REFERENCES

[1] C. McKay and I. Fujinaga, "Automatic music classification and the importance of instrument identification," in *Proceedings of the Conference on Interdisciplinary Musicology*, 2005.

[2] B. L. Sturm, M. Morvidone, and L. Daudet, "Musical instrument identification using multiscale mel-frequency cepstral coefficients," *Proc. of the European Signal Processing Conference (EUSIPCO)*, pp. 477–481, 2010.

[3] R. Rui and C. Bao, "Projective non-negative matrix factorization with Bregman divergence for musical instrument classification," in *Signal Processing, Communication and Computing (ICSPCC), 2012 IEEE Int. Conf. on*, aug. 2012, pp. 415 –418.

[4] P. J. Moreno and S. Agarwal, "An experimental study of EM-based algorithms for semi-supervised learning in audio classification," in *Proc. of the ICML-2003 Workshop on the Continuum from Labeled to Unlabeled Data*, 2003.

[5] J. H. Jeon and Y. Liu, "Semi-supervised learning for automatic prosodic event detection using co-training algorithm," in *Proc. of the Joint Conf. of the 47th Annual Meeting of the ACL and the 4th Int. Joint Conf. on Natural Language Processing of the AFNLP*, Suntec, Singapore, August 2009, pp. 540–548, Association for Computational Linguistics.

[6] Z. Zhang and B. Schuller, "Semi-supervised learning helps in sound event classification," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE Int. Conf. on*, March 2012, pp. 333–336.

[7] T. Li and M. Ogihara, "Semi-supervised learning for music artists style identification," in *Proc. of the thirteenth ACM Int. Conf. on Information and knowledge management*, New York, NY, USA, 2004, CIKM '04, pp. 152–153, ACM.

[8] Y. Song, C. Zhang, and S. Xiang, "Semi-supervised music genre classification," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE Int. Conf. on*. IEEE, 2007, vol. 2, pp. II–729.

[9] W. You and R. Dannenberg, "Polyphonic music note onset detection using semi-supervised learning," *Proc. ISMIR, Vienna, Austria*, 2007.

[10] S. Z. K. Khine, T. L. Nwe, and H. Li, "Singing voice detection in pop songs using co-training algorithm," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE Int. Conf. on*. IEEE, 2008, pp. 1629–1632.

[11] D. Little and B. Pardo, "Learning musical instruments from mixtures of audio with weak labels," in *ISMIR'08*, 2008, pp. 127–132.

[12] K. Jensen, *Timbre Models of Musical Sounds: From the Model of One Sound to the Model of One Instrument*, Report. Københavns Universitet, Datalogisk Institut, 1999.

[13] P. Cosi, G. De Poli, and P. Prandoni, "Timbre characterization with mel-cepstrum and neural nets," in *Proc. of the Int. Computer Music Conf. (ICMC)*, 1994, pp. 42–45.

[14] A. Eronen, "Comparison of features for musical instrument recognition," in *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2001.

[15] G. De Poli and P. Prandoni, "Sonological models for timbre characterization," *Journal of New Music Research*, vol. 26, no. 2, pp. 170–197, 1997.

[16] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. pp. 1–38, 1977.

[17] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Music genre database and musical instrument sound database," in *Proc. of the 4th Int. Conf. on Music Information Retrieval (ISMIR)*, 2003, pp. 229–230.