Co-Training for Classification of Live or Studio Music Recordings

Nicolas Auguin, Pascale Fung
Human Language Technology Center
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
{njlpauguin, pascale}@ust.hk

Abstract

The fast-spreading development of online streaming services has enabled people from all over the world to listen to music. However, it is not always straightforward for a given user to find the "right" song version he or she is looking for. As streaming services may be affected by the potential dissatisfaction among their customers, the quality of songs and the presence of tags (or labels) associated with songs returned to the users are very important. Thus, the need for precise and reliable metadata becomes paramount. In this work, we are particularly interested in distinguishing between live and studio versions of songs. Specifically, we tackle the problem in the case where very little-annotated training data are available, and demonstrate how an original co-training algorithm in a semi-supervised setting can alleviate the problem of data scarcity to successfully discriminate between live and studio music recordings.

Keywords: Music Information Retrieval, Co-training, Machine Learning

1. Introduction

Nowadays, music streaming services allow hundreds of millions of people across the world to access billions of songs. This has profoundly changed the experience of everyday music listeners. The listening experience of the users can be greatly enhanced by using recommendation tools, enabling the discovery of new songs, new genres or new emotions, that the user may not have had the opportunity to experience (or even think of) without the Internet. In order to fully exploit the vast resource that the Internet embodies, efficient search capabilities are essential. However, if data is cheap, information is expensive. Similarly to text-based information retrieval, music is prone to search errors resulting from noisy and potentially misleading metadata. Many online music databases, such as YouTube videos, are user-generated and therefore very mixed in terms of quality. In this work, we are particularly interested in learning from a music data set where songs are split into two classes: live and studio. We aim to determine if a random song taken from our corpus was recorded in a studio or if it is a live recording. Under a supervised setting, this is a typical, machine learning classification problem, which has previously been solved with very high accuracy (Auguin et al., 2013). However, in this paper, we focus on a case where only few labeled data are available. Under a semi-supervised setting, we propose to apply a co-training algorithm (Blum and Mitchell, 1998) to iteratively learn from our data by exploiting distinct views of our system. Co-training aims at combining labeled and unlabeled data to ultimately outperform a supervised system, which would typically perform poorly if only a small training set was at hand.

In this work, we design our own co-training framework and describe two effective settings under which we propose to use distinct views, in order to build a consistent and robust system under a semi-supervised setting. The rest of this paper is organized as follows: related work is discussed in section II. In section III, we introduce our training data and recall some of the results obtained under a supervised setting. In section IV, we review the co-training algorithm and propose our own design. We then present our methodology and our experimental setup in section V. We conclude the paper in section VI.

2. Related work

The co-training algorithm was introduced in a milestone paper by Blum and Mitchell (1998), in which the authors combine labeled and unlabeled data to improve the results of a supervised web-classification task. For this purpose, they partition each web page into two distinct views, namely, the words occurring on a specific page and the words occurring in hyperlinks that redirect to that page. Two learning algorithms are then trained on each view separately. Tested on new, unlabeled instances, each algorithm provides a prediction, based on which it is decided to add a given example to the originally-small labeled data set, thus enlarging the training data set at hand.

Co-training has already been used in music-related fields. In particular, this algorithm (with a Maximum Entropy model as machine learning framework) has been used in order to distinguish between speech and music (Wei et al., 2010). Unfortunately, though they report a high precision on this task, the improvement due to the use of co-training is not clearly expressed since no comparison is made when using only the initial annotated data (and thus ignoring the unlabeled data). Besides this, their co-training setting, especially the way their feature set is divided in distinct views, was not clearly presented. In music information retrieval, co-training has been used for genre classification (Xu et al., 2005), and for mood classification (Zhao et al., 2010). Xu et al. (2005) propose to split their audio feature set into two parts: features extracted using Fast Fourier Transform (FFT)-based computation form the first view, while Discrete Wavelet Transform (DWT)-based features form the second. On a three-class classification task, their co-training algorithm has been proved to outperform a variety of other classifiers. As for Zhao et al. (2010), they make full use of the intrinsic multi-modal nature of their data set by splitting their feature set into "natural" views: MIDIbased, audio-based and lyric-based features are exploited as three different views.

For our task of live/studio music data classification, we propose to build a solid co-training framework, within which supervised and semi-supervised settings will be compared. Unlike Zhao et al. (2010), we can not use lyric-based features as a view of our corpus, since the lyric-content of a song does not differ between studio and live-recorded versions of a song. Instead, we propose two different ways of building distinct views of our data set: the first way consists of splitting our audio feature set into supposedly independent and compatible views, while the second way exploits the diversity of different classifiers to provide different perspectives of the same feature subset.

3. Music corpus

3.1. Music data set

By using online music download services, we constituted a music data set composed of 1066 unique songs from various genres (rock, pop, jazz...) and in different languages (English, French, Spanish, Chinese, Portuguese). The class distribution can be seen in Table 1.

Live songs	Studio songs	Total songs
378	688	1066

Table 1: Class distribution of the music data set

The label “studio” refers to songs recorded in a studio, whereas the label “live” corresponds to songs recorded during live performances (e.g. concerts or public events). All songs were extracted from original albums, thereby providing us with the “ground truth”.

3.2. Feature set

From this data set, we extracted different subsets of features, namely Mel-Frequency Cepstral Coefficients (MFCCs), Linear Predictive Cepstral Coefficients (LPCCs), MPEG-7 features and psycho-acoustic features as well as beat histograms and signal energy-based features (Auguin et al., 2013). These features were extracted from 30-second samples of each song, after converting them to $22,050$ Hz and 16 bits format and mono channel PCM WAV files. The sample chosen for the feature extraction was the segment from 0 to 30 seconds for each song. This choice is not standard in Music Information Retrieval, but it has been proven to perform better than other segments of the song on this particular classification task (Auguin et al., 2013).

4. Classification using co-training

4.1. The co-training setting

The co-training setting (Blum and Mitchell, 1998; Nigam and Ghani, 2000) assumes that we have two (or more) independent and compatible views of our data. By independent, we mean that the views are actually uncorrelated, and by compatible, we mean that a classifier trained on either view will lead to the same classification in a given instance.

4.2. The co-training algorithm

The co-training algorithm is a weakly supervised machine learning algorithm which aims at exploiting the supposedly intrinsic independence of the views, under a co-training setting. Classifiers are trained on the small, initial labeled data set using either feature set. They are then tested on the whole remaining set, assigning to each test instance a class label (in our case, studio or live). The instances labeled with the highest confidence are then added to the labeled data set, and this is done for all views. The process is repeated until all instances are labeled. The pseudo-code of the co-training algorithm is given below.

```
Data: A set L of labeled training instances and a set U of
      unlabeled instances
Result: Loop for n iterations:
while U is not empty do
    For each view, train each classifier on L;
    Add to L the instances of U labeled with the highest
    confidence by each classifier;
end
```

Algorithm 1: The co-training algorithm

As no human annotation is required in the whole classification process, this algorithm constitutes a powerful alternative to other semi-supervised schemes, such as active learning, which still involves human effort.

5. Methodology

In this paper, we propose a variant of the co-training algorithm where at each stage, instances labeled with a confidence prediction higher than a certain threshold are added to the pool. This confidence threshold (between 0 and 1) is lowered (e.g., by 0.02) at each iteration. This implies that the more reliable instances are typically added to the pool in the early stages of the co-training, whereas the less reliable are added at the end of the co-training process. Hence we have the following pseudo-code:

```
Data: A set L of labeled training instances and a set U of
      unlabeled instances
while Threshold $T > 0$ do
    For each view, train each classifier on L;
    Add to L the instances of U labeled with a confidence
    $> T$ by each classifier;
end
```

Algorithm 2: Our proposed variant of the co-training algorithm

5.1. Multi-view feature set

We propose to build two different views of our data set by considering separately MFCCs (whose concatenation results in a 39-dimensional feature vector) and beat histograms features (an 18-dimensional feature vector). This artificial split rests on the two following observations: first, from an extraction point of view, MFCCs are extracted using FFT, whereas beat histograms are computed using DWT. Second, from a music perspective, MFCCs correspond to low-level features, while beat histograms correspond to mid-level features. Therefore, we can assume that both these views are independent. We also assume that both views are compatible, in other words, they give the same class label to each example.

5.1.1. Experimental set-up

In order to evaluate the performance of co-training w.r.t. to a supervised scheme, we first perform a 10-fold stratified split of our data set. Therefore, 90% of the data are used for training/co-training, while the other 10% are used only for testing. The stratification ensures that class probabilities remain roughly the same within each fold. This is a typical 10-fold separate process.

From the 90% set, n instances are randomly chosen to constitute the initial (labeled) pool. The remaining instances are assumed to be unknown (no annotation provided). Our variant of the co-training algorithm is then used, combining both the initial pool and unlabeled data. Ultimately, as the confidence threshold is lowered at each iteration, the pool issued from the co-training process is larger than at the starting point. A classifier is then trained on this pool and tested on the 10^{th} fold. The same classifier is trained on the original pool and tested on the same fold, which enables us to compare a supervised scheme (with small training data) with our proposed co-training framework.

The underlying classifier used in this part is Support Vector Machine with linear kernel, with probabilistic outputs (Platt, 1999). Trained on both feature subsets, the resulting predictions for each view are combined using a sum rule, which outputs one final confidence prediction for each instance. This prediction is then compared to the confidence threshold introduced above. The instance is finally chosen either to be added to the pool or not. In the following results, we consider the average over the 10-folds. Before the co-training process, instances are randomly picked to constitute the initial pool. Therefore, the results presented in the following sections are the average over 5 random realizations.

We study the performance of these two schemes at different stages of the co-training process, i.e., for different sizes of the pool built using the co-training algorithm, when only 15 examples are initially labeled.

5.1.2. Results

The results can be seen in Table 2. We also provide the learning curve of the co-training algorithm in Fig 1.

We can see that under a supervised setting, with only 15 initially-labeled examples, SVM performs at 69.7% on average on the test set, whereas SVM after co-training leads to a 79.1% performance after 50 iterations, resulting in a 10% global improvement. This shows that intelligent use of the unlabeled data set has boosted our initial system.

5.2. Multi-view system using various classifiers

In the previous section, we assumed that the two views corresponding to MFCC-based features and beat histogram-based features were independent. As an alternative, in this section we propose to use different classifiers on the same feature subset, expecting that distinct machine learning methods may learn different perspectives of our data. In this multi-ensemble learning framework, we assume again that classifiers' views are independent and compatible, in order not to violate the co-training setting.

Confidence threshold/ Iteration number	Average size of the labeled pool	Global accuracy	Learning scheme
1 / **0**	15	**69.7%**	Supervised setting
0.94 / 3	188.2	70.6%	Co-training
0.84 / 8	588.2	76.1%	Co-training
0.70 / 15	745.2	77.6%	Co-training
0.40 / 30	822.6	75.9%	Co-training
0 / **50**	954	**79.1%**	Co-training

Table 2: Global accuracy performed using a multi-view feature set and 15 initial annotations

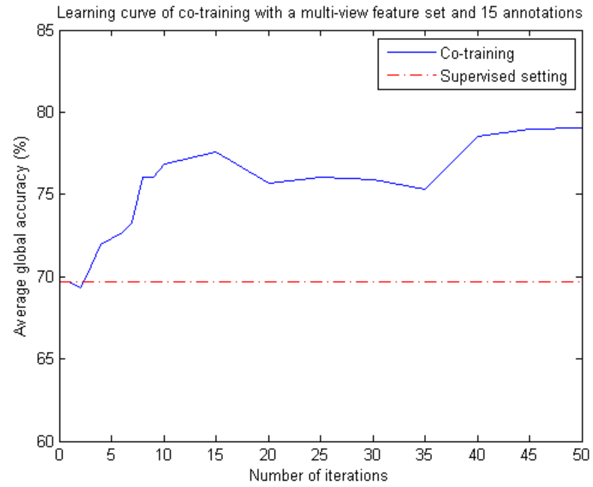

Figure 1: Learning curve of co-training with a multi-view feature set

5.2.1. Experimental set-up

The experimental set-up is the same as in the previous section. However, we now consider only MFCCs as features, resulting in a 39-dimensional feature vector. We propose to train three different classifiers on this feature set: SVM with linear kernel, Decision Trees, and Naive Bayes.

As before, for each unlabeled instance, predictions of each classifier are combined using a sum rule, before the final decision (instance to be added to the pool or not) is made.

5.2.2. Results

Results can be seen in Table 3. The learning curve of the co-training algorithm is also provided in Fig 2.

Once again, we observe that combining labeled and unlabeled data leads to much better performance than using labeled data alone. However, unlike in the previous section, we can see that the performance does not increase as the pool is expanded. For example, after 15 iterations, SVM

Confidence threshold/ Iteration number	Average size of the labeled pool	Global accuracy	Learning scheme
1 / **0**	15	**69.6%**	Supervised setting
0.94 / 3	734.2	78.5%	Co-training
0.84 / 8	797.4	79.8%	Co-training
0.70 / **15**	844.2	**80.9%**	Co-training
0.40 / 30	899.6	78.9%	Co-training
0 / 50	958.2	77.8%	Co-training

Table 3: Global accuracy performed using different classifiers' views and 15 initial annotations

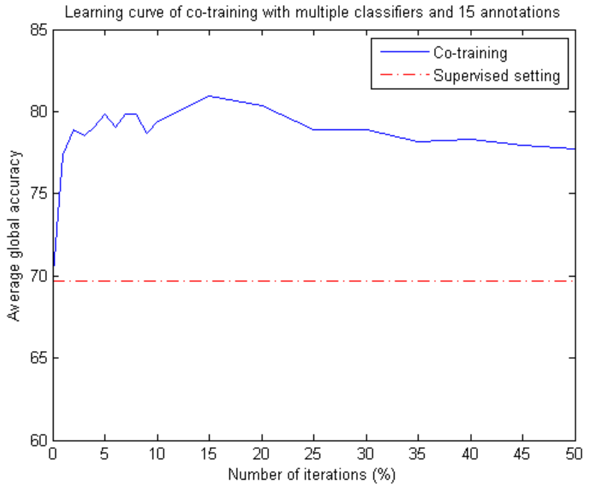

Figure 2: Learning curve of co-training using multiple classifiers

trained on roughly 88% of the data leads to an 80.9%-accuracy, whereas after 50 iterations, the global accuracy reaches 77.8%. This implies that after a number of iterations, classifiers tend to misclassify the same examples, leading to a decrease in accuracy.
On the other hand, we can note that only a few iterations (e.g., 15) are needed to lead to high accuracy (e.g.; 80.9%) in the multi-classifier setting, while the multi-feature set typically needs 50 iterations to reach similar accuracy (79.1%).

6. Conclusion

We propose a novel semi-supervised solution to the problem of classifying studio and live-recorded versions of songs. This work provides insightful perspectives on information retrieval, when many unlabeled data are available, but only limited labeling information is at hand. The co-training algorithm is proven to be both robust and efficient (only 15 labeled instances can lead to a global accuracy up to 80%) and offers a powerful way to alleviate manual annotation or tagging. Thus, it can be of great help for constituting resources for online streaming platforms. In future work, we plan to apply the proposed co-training algorithm to other music-related classification tasks, for example audio event analysis within music.

7. References

Auguin, N., Huang, S., and Fung, P. (2013). Identification of live or studio versions of a song via supervised learning. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, pages 1–4. IEEE.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM.

Nigam, K. and Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 86–93. ACM.

Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Wei, Z., Qun, Z., Yayu, L., and Minhui, P. (2010). Co-training approach for label-minimized audio classification. In *Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on*, volume 1, pages 860–863. IEEE.

Xu, Y., Zhang, C., and Yang, J. (2005). Semi-supervised classification of musical genre using multi-view features. In *International Computer Music Conference (ICMC 2005)*, pages 5–9.

Zhao, Y., Yang, D., and Chen, X. (2010). Multi-modal music mood classification using co-training. In *Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on*, pages 1–4. IEEE.