ABSTRACT

Separating singing voice from a polyphonic mixed audio signal is a challenging but important task, with a wide range of applications across the music industry and music informatics research. Various methods have been devised over the years, ranging from Deep Learning approaches to dedicated ad hoc solutions. Here, we present a novel machine learning method for this task, using a Conditional Random Field (CRF) approach for structured output prediction. We exploit the diversity of previously proposed approaches in using their predictions as input features to our method – thus effectively becoming an ensemble method. Our empirical results demonstrate the potential of integrating predictions from previously proposed different methods into one ensemble method, and additionally show that CRF models with larger complexities generally lead to superior performance.

Index Terms — Singing voice separation, conditional random fields, ensemble method

1. INTRODUCTION

1.1. Background

Singing Voice Separation (SVS) is the task of deconstructing an audio mixture containing several sources into two components: the sung melody (the vocals) and everything else (the background). The problem is commonly approached in the time-frequency domain. First, a spectrogram is computed using the Short-Time Fourier Transform (STFT) applied to the mixed audio signal. The resulting spectrogram image is a matrix where horizontal axis represents time, the vertical axis frequency and the amplitude of a particular (time, frequency) pair is the intensity of the corresponding pixel in the image. Then, a typical SVS algorithm will classify each pixel in the spectrogram as belonging to either the vocals or the background. This results in a binary/hard mask: a matrix of the same dimensions as the spectrogram which contains a 1 whenever the energy at the corresponding pixel is deemed to be due to vocals, and a 0 otherwise. Some methods take a less rigid approach and determine for each pixel the proportion of the energy at the corresponding time and frequency that is ascribed to the vocals and to the background. Such methods result in a continuous/soft mask, which contains values in the range [0, 1] representing the predicted proportions, rather than binary values [1]. Given either type of mask, it is then possible to reconstruct the time-domain signal of both vocal and background, simply by element-wise multiplying the spectrogram with either the mask (for the vocals) or 1 minus the mask (for the background) and computing the inverse STFT of the result.

It is important to point out that the masking approach is potentially imperfect and may not yield optimal SVS results. Indeed, metrics used for evaluating the quality of the output of an SVS approach are complex and do not rely on the masking assumption, such that even the true mask may be imperfect according to these metrics.

1.2. Machine learning approaches to SVS

Several recent methods have adopted a machine learning approach in order to train the algorithm for predicting the mask. Such strategies require a training dataset containing the spectrogram along with the ground-truth mask for a sufficiently large set of songs. However, creating such annotations is clearly non-trivial and extremely challenging to do by hand.

Recently researchers have created an automated approach for extracting such a ground-truth mask, referred to as the ‘Ideal Binary
1.3. Paper structure

The remainder of this paper is arranged as follows. In Section 2 we review the literature relevant to SVS. Section 3 introduces our models, which are evaluated in Section 4. We conclude in Section 5.

2. RELATED WORK

Many existing approaches for SVS are based on matrix decomposition techniques applied to magnitude spectrograms. Examples include Independent Component Analysis [3], Robust Principal Component Analysis [4], harmonic-percussive source separation [5] and dictionary learning [6]. In contrast to these methods, an alternative approach is to track the sung melody more directly, by estimating the \( f_0 \) (fundamental frequency) of the estimated vocal melody, and re-constructing a binary mask from the \( f_0 \) trajectory as well as a number of its harmonics [7]. Also related to our work is the research by Lagrange et al. [8], who use a graph cutting algorithm to divide a binary mask into vocal and non-vocal segments. Recent developments in this area include Deep Learning approaches [9, 10], online real-time methods [11] and the REPET system [12].

The publication of publicly-available datasets such as the MIR1k dataset [13] and iKala dataset [14] have also helped benchmark algorithms. The authors of the iKala dataset also held back a set of songs for testing within the MIREX (Music Information Retrieval Evaluation eXchange) 2014 Singing Voice Separation task\(^1\), which featured 11 algorithms from 8 different teams.

The use of Conditional Random Fields to model music is not novel as such. CRFs are a powerful probabilistic framework, particularly well-suited to music information retrieval as they can effectively learn from sequential data. They determine a mapping from a sequence of feature vectors, including overlapping and non-independent features, to a sequence of labels. CRFs have been successfully applied to several music informatics tasks such beat tracking [15], audio-to-score alignment [16] or modelling musical emotions over time [17].

Conditional Random Fields (CRFs) have been used extensively in particular in machine vision applications (e.g. [18]), and also in other areas of audio, speech, language and music analysis [19, 20, 21, 22]. Given the similarity between visual object recognition and SVS, they are thus a natural choice.\(^2\) With respect to ensemble methods, Le Roux et al. [23] used a similar approach to our own in the related task of speech enhancement.

3. PROPOSED METHODS

The approach proposed in this paper is based on binary mask learning. In particular, we seek a function which maps the spectrogram of a mixed music audio signal to a binary mask, which labels each pixel in the spectrogram as 1 (vocal) or as 0 (non-vocal).

The baseline approach we propose is to simply binary classify each pixel in the spectrogram, based on a feature representation of the pixel. However, this disregards the dependencies between neighbouring pixels. To exploit the dependencies, we also investigate structured output prediction approaches which do not aim to predict the label of each pixel in isolation, but the entire binary mask (or large chunks of it) at once. Depending on the dependencies considered, this leads to models of varying complexities.

We begin this section with a description of the features we computed for a spectrogram, which will remain constant over all experiments conducted within this work. Then we will present the different classification approaches considered.

3.1. Feature extraction

Let \( \mathbf{X} \in \mathbb{R}_+^{F \times T} \) represent a magnitude spectrogram for an audio mixture with \( F \) frequency bins and \( T \) time frames. From this spectrogram, we computed different features that we considered of potential relevance to the task at hand.

**Sparse component of Robust PCA** Robust Principal Component Analysis [24, 4] is a variant of PCA which aims to decompose the matrix \( \mathbf{X} \) into a low-rank term and a sparse term \( \mathbf{S} \in \mathbb{R}_+^{F \times T} \). As the singing voice is less regular and more sparsely present in the spectrogram than the instrumental accompaniment, we included the sparse component (setting the \( \ell_1 \) weight penalty equal to the default \( 1/\sqrt{\max(F,T)} \)) as a feature. **Harmonic component** We split \( \mathbf{X} \) into its harmonic and percussive components using a median filtering approach [25], keeping the harmonic component \( \mathbf{H} \in \mathbb{R}_+^{F \times T} \) as a feature. **Gabor filtered spectrogram** Inspired by image processing where they have proven useful in a variety of tasks, we also included 4 Gabor filtered spectrograms [26] as features. The filters had rotation equal to 0, \( \pi/4 \), \( 2\pi/4 \) and \( 3\pi/4 \) and each had horizontal bandwidth equal to 1 and vertical bandwidth equal to 3 (empirically selected to attain a reasonable output on musical spectrograms). **The log power of the pixel** \( \log_{50} (\mathbf{X}(f,t)) \), as the power can be expected to be higher where the sung voice is (in logarithm scale to mimic the human auditory system). **The frequency \( f \) of the pixel itself** as the vocal activity has clear frequency biases.

These features have either directly been used in SVS or related tasks. However, an additional set of features, which are no doubt

\(^1\)http://www.music-ir.org/mirex/wiki/2014:Singing_Voice_Separation

\(^2\)Note that the term CRF is used slightly abusively in this community, for structured output prediction methods that model just pairwise dependencies between atomic labels; we adopt the same abuse of terminology—in fact our CRF methods are trained using a maximum margin approach.
informative, are the per-pixel predictions of existing SVS algorithms. We therefore included the predictions of two state-of-the-art and complementary existing systems on X as two extra features (both of which output a soft mask the same dimensions as X): REPET REPeating Pattern Extraction Technique [12]. The Deep Learning system for SVS [27]. The features above were finally concatenated into a 10-dimensional feature vector (S, H, 4 Gabor filter outputs, log power, frequency, REPET output, Deep Learning output).

### 3.2.2. Modelling time dependencies

Vocal activity within a spectrogram is likely to be non-stationary, meaning that we may gain performance by allowing time-adjacent pixels to affect the likelihood that a certain pixel contains vocal energy. Thus, we trained a CRF model in which the hidden nodes correspond to the elements in the binary mask, and the hidden graph structure over these nodes consists of a set of chains across time, one for each frequency. Edges within the model were specified to be undirected and learning was then conducted using the block co-ordinate Frank-Wolfe algorithm [28]. We refer to this model as Time.

The regularisation parameter C was roughly tuned on a small subset of the data, after which it was set to 10\(^{-5}\) across all experiments – further optimisation using cross-validation is computationally very challenging but may yield improvements in performance.

### 3.2.3. Modelling frequency dependencies

With similar motivation to above (vocal frequencies will typically occupy more than one frequency band in a spectrogram), we also trained a CRF with dependencies between nodes representing frequency-adjacent mask elements. The model was set up in exactly the same way as 3.2.2. We refer to this model as Frequency.

#### 3.2.4. Modelling both time and frequency dependencies

A natural extension of the models above is to model the horizontal axis (time) and vertical axis (frequency) dependencies simultaneously. This flexibility gives CRFs a distinct advantage over simpler graphical models such as Hidden Markov Models. The graph structure in this model was set such that each node was connected to its immediate neighbours above, below, to the left and right.

Although exact inference methods are known for grid models such as these, we found that they were too computationally expensive for our purpose. We therefore used the same approximate learning method as in the two previous methods. We refer to this model as 4-connected.

### 4. EXPERIMENTS

#### 4.1. Description of the dataset

The dataset for this work consisted of the publicly-available subset of the iKala dataset, which consists of 252 30-second clips of Chinese pop/rock music\(^3\). Each audio example contains two channels: one containing the vocals and the other containing the background. A magnitude spectrogram for each of these channels was computed, resulting in two spectrograms, \(V, B \in \mathbb{R}_+^{F \times T}\). An ideal binary mask was then computed via element-wise comparison of \(V\) with \(B\):

\[
\text{IBM}_{f,t} = \begin{cases} 
1 & \text{if } V_{f,t} > B_{f,t} \\
0 & \text{otherwise.}
\end{cases}
\]

These masks were then used as the ground truth labels for training our method, as well as giving an upper bound on the performance.

#### 4.2. Setup of the experiments

To ensure compatibility with the Deep Learning method (which, recall, is one of our features), audio was downsampled to 16kHz. The loudness of the vocals in each song was set to be equal (0dB) to the background. Spectrograms were computed with a window length of 1024 samples with a hop of 256 frames. Audio processing was conducted using librosa [29] and sci-kit image [30] (for the Gabor filters),

---

\(^3\)http://www.zafarrafii.com/codes/repet_sim.m

\(^4\)https://github.com/posenhuang/deeplearningsourceseparation

\(^5\)http://mac.citi.sinica.edu.tw/ikala/
classification was performed using scikit-learn [31] and PyStruct [32], and evaluation was performed using the BSS-toolbox [33]. Audio was upsampled back to the native 44kHz before conducting evaluation to avoid any signal processing artifacts affecting the performance.

Evaluation was conducted using 10-fold cross validation, with 25 of the 252 songs held out for testing in each fold. Unfortunately, memory constraints made it impossible to make use of the full remaining 90% of songs for training in each fold: we decided to sample just 25 songs at random from the training set (note that the same random set was used across all different methods). This means that the reported performances of the newly proposed methods are likely to be underestimates of what can be achieved using more working memory, or with a parallel implementation (which is subject of our future work).

4.3. Results and discussion

The most common metrics for evaluating blind audio source separation methods (of which SVS can be considered a subfield) are the SDR (Signal to Distortion Ratio), SIR (Source to Interference Ratio), and SAR (Source to Artifacts Ratio) [33]. We used these metrics to measure the efficaciousness of our methods, accounting for the levels in the true mix as suggested by the MIREX team. Results are shown in Table 1, where in addition to the four proposed methods, we also show the performance of REPET and the Deep Learning system, as well as the performance of the ideal binary mask.

The first two rows in Table 1 represent existing systems, which attain between 3.72 and 7.91dB NSDR for the sung voice and up to 5.75dB for the musical accompaniment. The remaining rows are ordered in terms of increasing model complexity. Our methods in general offer an improvement in terms of NSDR, with the more complex models (Time, Frequency, 4-connected) achieving superior performance, although further improvements could be expected for the more complex models if we would have been able to utilize more of our training data. Our models also perform well with respect to SIR, especially on the voice. In terms of SAR, we fall short of the score attained by REPET, which could be due to the binary mask introducing artifacts.

An example of the output of our system is shown in Figure 2. Note that although the output for REPET and our proposed 4-connected method appears similar, in this example we achieved a increase in NSDR of more than 7dB for the sung voice over REPET. We refer the reader to our website for audio examples.

A statistical analysis of the methods revealed that, although the magnitude of improvement across methods is small, in most cases it was significant to a high level. In particular our best-performing method in terms of NSDR on the sung voice, 4-connected, was a significant improvement over all other methods, with $p$-values all below $10^{-4}$ using the Wilcoxon signed-rank test.

4.4. MIREX Evaluation

To evaluate our methods more directly against cutting edge systems, we also submitted our algorithm to the 2015 MIREX SVS task, which contained audio clips unavailable to participants. In this setting, our algorithm slightly underperformed compared to systems by more experienced teams. However, no algorithm outperformed any other when variance across test songs was taken into account.

### Acknowledgements

This work was funded by EPSRC grant EP/M000060/1. We would like to thank the authors of the REPET and Deep Learning system for making their code available online.
6. REFERENCES


