LEARNING TO SEPARATE VOCALS FROM POLYPHONIC MIXTURES VIA ENSEMBLE METHODS AND STRUCTURED OUTPUT PREDICTION

M. McVicar, R. Santos-Rodríguez, T. De Bie

Intelligent Systems Laboratory
Department of Engineering Mathematics
University of Bristol

ABSTRACT

Separating the singing 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. In this paper, we present a novel machine learning method for the task, using a Conditional Random Field (CRF) approach for structured output prediction. We exploit the diversity of previously proposed approaches by using their predictions as input features to our method – thus effectively developing an ensemble method. Our empirical results demonstrate the potential of integrating predictions from different previously-proposed 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 task is commonly approached in the time-frequency domain. First, a *spectrogram* is computed using the Short-Time Fourier Transform (STFT) of the mixed audio signal. The resulting spectrogram image is a matrix where the horizontal axis represents time, the vertical axis represents frequency and the amplitude of a particular (time, frequency) pair is indicated by 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 when it is deemed to be due to the background. 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 vocals 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.

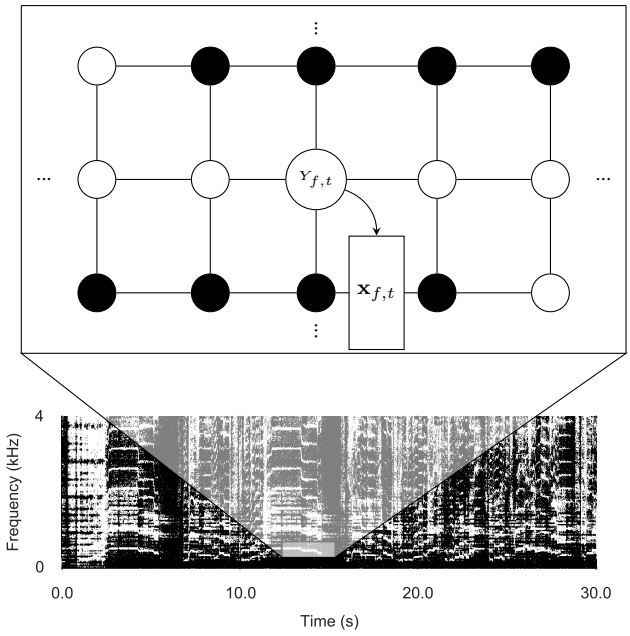

Fig. 1. Conditional Random Fields for singing voice separation. A feature vector $\mathbf{x}_{f,t}$ is computed for each pixel in a spectrogram. This alone can be used to classify $Y_{f,t}$ as either vocal (shown here in white) or non-vocal (black). However, we may also make use of the neighbours of $Y_{f,t}$ to assist with prediction.

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 a ground-truth mask, referred to as the 'Ideal Binary Mask'

(IBM)[2], based on the separate spectrograms of the vocals and the background. This approach takes the element-wise maximum of the magnitude spectrum of vocal and background audio tracks. Besides their use in training, IBMs are also useful for evaluating on hold-out sets or using cross-validation techniques. Furthermore, they are useful in upper bounding the performance of any masking approach.

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 reconstructing a binary mask from the f_0 trajectory alongside 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. The most recent developments in this area include Deep Learning approaches [9, 10], online real-time methods [11] and the REPET system [12].

The recent publication of publicly-available datasets such as the MIR1k dataset [13] and iKala dataset [14], have also helped benchmark algorithms. For example, 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 (CRFs) 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] and the modelling of musical emotions over time [17].

Conditional Random Field approaches have been used particularly extensively in machine vision applications (e.g. [18]), as well as 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 when addressing the related task of speech enhancement.

2.1. Contributions

In this paper, we present a sequence of models of increasing complexity which aim to predict a binary mask given a spectrogram. Underlying each of these methods is the idea that each pixel is associated not with a single value but with a set of features collected in a feature vector. These sets range from simple low-level features of the spectrogram to high-level features, such as the values of the predicted masks developed using previously-propsed methods. As such, our proposal effectively represents a type of ensemble method.

Our baseline model classifies each pixel independently using logistic regression. This simple approach does not exploit dependencies between nearby pixels in the mask. Indeed, vocal activity is likely to vary slowly over time (far slower than the frame rate of a spectrogram), and is not likely to occupy a single frequency band (see Figure 1). To exploit this, our subsequent models make use of Conditional Random Fields to encode our assumptions: the first model includes dependencies between time-adjacent entries of the mask, while the second model considers frequency-adjacent nodes. Finally, in the third model, both dependencies are accounted for.

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 (background).

The baseline approach we propose is to simply classify each pixel in the spectrogram into vocal or background, based on a feature representation of the pixel. This approach disregards the dependencies between neighbouring pixels so to exploit these dependencies we investigate structured output prediction approaches which do not aim to predict the label of each pixel in isolation, but rather aim to predict the entire binary mask (or large chunks of it) at once. Models of varying complexities are therefore developed, according to the dependencies considered.

We begin this section with a description of the features we computed for a spectrogram, which will remain constant over all experiments outlined in this paper. We will then 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 our task.

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 L_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 $0, \pi/4, \pi/2$ 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** $\log_{10}(\mathbf{X}(f,t))$, as the power can be expected to be higher where the sung voice is present (in logarithmic scale to mimic the human auditory system). **The frequency f of the pixel itself** as the vocal activity has clear frequency biases.

[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.

Method	NSDR		SIR		SAR	
	Voice	Music	Voice	Music	Voice	Music
REPET	7.91 ± 3.30	5.78 ± 3.49	8.36 ± 9.25	15.59 ± 5.14	$\mathbf{9.34 \pm 2.62}$	$\mathbf{9.38 \pm 2.68}$
Deep	3.72 ± 1.35	-0.04 ± 5.23	1.62 ± 5.86	$\mathbf{18.75 \pm 4.95}$	7.98 ± 2.84	7.97 ± 2.84
Independent	7.08 ± 2.59	3.86 ± 4.41	9.59 ± 8.57	18.28 ± 5.00	6.15 ± 3.58	6.19 ± 3.63
Time	9.26 ± 3.64	5.80 ± 3.53	17.21 ± 9.65	16.46 ± 5.58	6.51 ± 3.43	6.55 ± 3.47
Frequency	9.16 ± 3.62	5.71 ± 3.54	16.95 ± 9.62	16.44 ± 5.60	6.44 ± 3.44	6.47 ± 3.47
4-connected	$\mathbf{9.30 \pm 3.62}$	$\mathbf{5.82 \pm 3.45}$	17.19 ± 9.54	16.12 ± 5.54	6.54 ± 3.41	6.58 ± 3.46
Ideal Binary Mask	17.14 ± 3.39	12.80 ± 3.63	31.21 ± 3.70	27.50 ± 3.93	13.33 ± 3.48	13.37 ± 3.51

Table 1. Normalised Source to Distortion Ratio (NSDR), Source to Interferences Ratio (SIR), Sources to Artifacts Ratio (SAR) for our experiments. All results are measured in dB relative to the true mix and show mean and standard deviation of performance over all test songs. Best results in each column are shown in boldface.

These features have been used either directly in SVS or in related tasks. An additional set of features which are also no doubt 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 $\mathbf{X}$ as two extra features (both of which output a soft mask the same dimensions as $\mathbf{X}$): **REPET** REpeating Pattern Extraction Technique [12][3] and **The Deep Learning system for SVS** [27][4]. The features above were finally concatenated into a $10-$dimensional feature vector ($\mathbf{S}$, $\mathbf{H}$, 4 Gabor filter outputs, log power, frequency, REPET output, Deep Learning output).

3.2. Classification techniques

3.2.1. Independent model

Our first approach is simply to learn a logistic regression model from the feature space to $\{0, 1\}$. An L_2 norm penalty was specified with the intercept additionally fitted. In the test phase, each pixel in the spectrograms was then predicted independently, leading us to refer to this method as `Independent`.

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^{-7} 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 the above `Time` model (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. This model was set up in exactly the same way as 3.2.2 and we refer to it 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 the 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 - `Time` and `Frequency`. We refer to this final 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[5]. 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, $\mathbf{V}, \mathbf{B} \in \mathbb{R}_+^{F \times T}$. An ideal binary mask was then computed via element-wise comparison of $\mathbf{V}$ with $\mathbf{B}$:

$$\mathbf{IBM}_{f,t} = \begin{cases} 1 & \text{if } \mathbf{V}_{f,t} > \mathbf{B}_{f,t} \\ 0 & \text{otherwise.} \end{cases}$$

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 (one of our features as outlined earlier), 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). Classification was performed using scikit-learn [31] and

[3] `http://www.zafarrafii.com/codes/repet_sim.m`

[4] `https://github.com/posenhuang/deeplearningsourceseparation`

[5] `http://mac.citi.sinica.edu.tw/ikala/`

PyStruct [32]. Evaluation was performed using the BSS-toolbox [33]. Audio was upsampled back to the native 44kHz before evaluation to avoid interefereence from any signal processing artifacts.

Evaluation was conducted using $10-$fold cross validation, with 25 of the 252 songs in each fold held out for testing. Unfortunately, memory constraints made it impossible to make use of the full remaining 90% of songs for training in each fold: we therefore 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 will be the 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[6]. 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 5.78dB for the musical accompaniment. The remaining rows are ordered in terms of increasing model complexity. In general, our methods offer an improvement in terms of NSDR, with the more complex models (`Time`, `Frequency`, `4-connected`) achieving superior performance. Further improvements could be expected for the more complex models if we were able to use more of our training data. Our models also perform well with respect to SIR, especially in relation to the voice. In terms of SAR, we fall short of the score attained by `REPET` - this could a consequence of the binary mask introducing artifacts.

An example of the output of our system is shown in Figure 2. Note that although the outputs for REPET and our proposed `4-connected` model appear 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 [7].

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 other teams. However, no algorithm outperformed any other when variance across test songs was taken into account.

[6] `http://www.music-ir.org/mirex/wiki/2015:Singing_Voice_Separation#Evaluation`

[7] `http://www.interesting-patterns.net/ds4dems/vocal-source-separation/`

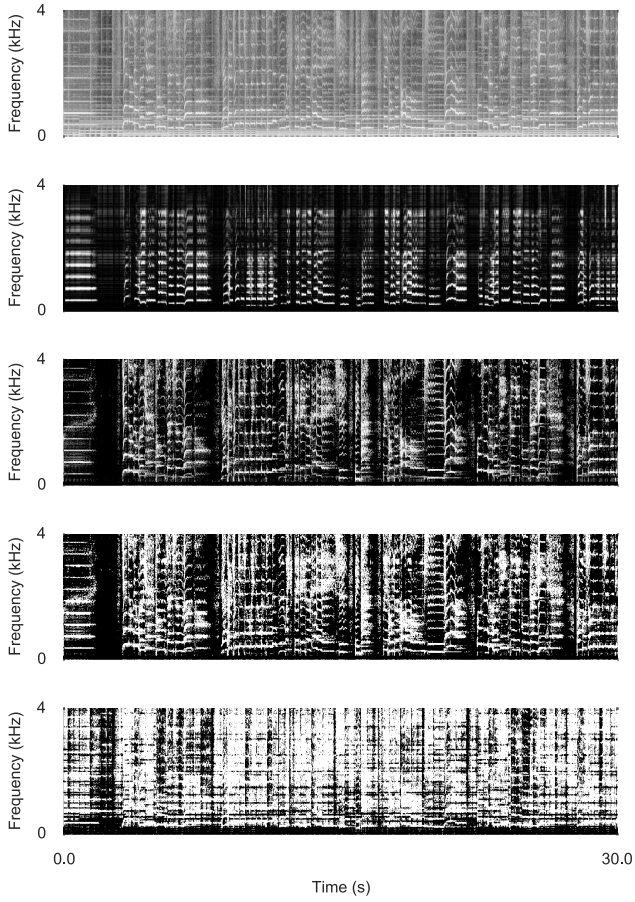

Fig. 2. Example output from our system. From top to bottom: log power spectrogram of mixture, Deep Learning system, REPET system, our proposed method (`4-connected` model), Ideal Binary Mask. In all images white indicates high energy.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an ensemble method for Singing Voice Separation. Using a combination of simple low-level features, matrix decomposition techniques, and the output of existing systems, we learnt a hard mask from the feature space to a {vocal, non-vocal} label space. Experimenting on publicly-available data, we achieved an increase of ~ 1.25dB NSDR relative to an existing methods. In terms of SIR, we made a larger gain of almost 9dB. Our algorithm was also submitted to the Music Information Evaluation eXchgange for evaluation against competing methods on held-out test audio.

For future work we would like to try an 8-connected grid (including diagonal neighbours), and investigate if more scalable methods would allow us to exploit more of the available training data. Other interesting potential avenues of future research include adding links between *harmonically* related nodes, and thoroughly investigating the relevance of the individual features we used.

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 and Maddy Wall for her proofreading.

6. REFERENCES

[1] Angkana Chanrungutai and Chotirat Ann Ratanamahatana, "Singing voice separation for mono-channel music using non-negative matrix factorization," in *Advanced Technologies for Communications, 2008. ATC 2008. International Conference on*. IEEE, 2008, pp. 243–246.

[2] G. Hu and D.L. Wang, "Monaural speech segregation based on pitch tracking and amplitude modulation," *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1135–1150, 2004.

[3] S. Vembu and S. Baumann, "Separation of vocals from polyphonic audio recordings," in *ISMIR*, 2005, pp. 337–344.

[4] P. Huang, S. Chen, P. Smaragdis, and M. Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *Acoustics, Speech and Signal Processing, IEEE International Conference on*, 2012, pp. 57–60.

[5] I. Jeong and K. Lee, "Vocal separation from monaural music using temporal/spectral continuity and sparsity constraints," *Signal Processing Letters, IEEE*, vol. 21, no. 10, pp. 1197–1200, 2014.

[6] Y.H. Yang, "Low-rank representation of both singing voice and music accompaniment via learned dictionaries.," in *ISMIR*, 2013, pp. 427–432.

[7] Y. Li and D.L. Wang, "Singing voice separation from monaural recordings," in *ISMIR*, 2006, pp. 176–179.

[8] M. Lagrange, L.H. Martins, J. Murdoch, and G. Tzanetakis, "Normalized cuts for predominant melodic source separation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 2, pp. 278–290, 2008.

[9] P. S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Singing-voice separation from monaural recordings using deep recurrent neural networks," in *ISMIR*, 2014.

[10] A.J.R. Simpson, G. Roma, and M.D. Plumbley, "Deep karaoke: Extracting vocals using a convolutional deep neural network," *arxiv. org abs/1504.04658*, 2015.

[11] P. Sprechmann, A.M. Bronstein, and G. Sapiro, "Real-time online singing voice separation from monaural recordings using robust low-rank modeling," in *ISMIR*, 2012, pp. 67–72.

[12] Z. Rafii and B. Pardo, "Repeating pattern extraction technique (repet): A simple method for music/voice separation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 1, pp. 73–84, 2013.

[13] C.L. Hsu and J.S.R. Jang, "On the improvement of singing voice separation for monaural recordings using the mir-1k dataset," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 2, pp. 310–319, 2010.

[14] T. Chan, T. Yeh, Z. Fan, H. Chen, L. Su, Y. Yang, and R. Jang, "Vocal activity informed singing voice separation with the ikala dataset," in *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing*, 2015, pp. 718–722.

[15] T. Fillon, C. Joder, S. Durand, and S. Essid, "A conditional random field system for beat tracking," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Brisbane, Australia, Apr. 2015.

[16] C. Joder, S. Essid, and G. Richard, "A conditional random field framework for robust and scalable audio-to-score matching," *IEEE Transaction on Audio, Speech and Language Processing*, vol. 19, no. 8, pp. 2385–2397, Nov. 2011.

[17] E.M. Schmidt and Y.E. Kim, "Modeling musical emotion dynamics with conditional random fields," in *ISMIR*, Miami (Florida), USA, October 24-28 2011, pp. 777–782.

[18] Nils Plath, Marc Toussaint, and Shinichi Nakajima, "Multi-class image segmentation using conditional random fields and global classification," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 817–824.

[19] E. Fosler-Lussier, Y. He, P. Jyothi, and R. Prabhavalkar, "Conditional random fields in speech, audio, and language processing," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1054–1075, 2013.

[20] Slim Essid, "A tutorial on conditional random fields with applications to music analysis," `http://perso.telecom-paristech.fr/~essid/teach/CRF_tutorial_ISMIR-2013.pdf`, 2013, Accessed: 2016-01-05.

[21] R. Prabhavalkar, Z. Jin, and E. Fosler-Lussier, "Monaural segregation of voiced speech using discriminative random fields," in *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[22] J. Woodruff, R. Prabhavalkar, E. Fosler-Lussier, and D. Wang, "Combining monaural and binaural evidence for reverberant speech segregation.," in *INTERSPEECH*. Citeseer, 2010, pp. 406–409.

[23] Jonathan Le Roux, Shigetaka Watanabe, and John R Hershey, "Ensemble learning for speech enhancement," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*. IEEE, 2013, pp. 1–4.

[24] E.J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 11, 2011.

[25] D. Fitzgerald, "Harmonic/percussive separation using median filtering," in *13th International Conference on Digital Audio Effects*, 2010.

[26] A.K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," in *Systems, Man and Cybernetics, 1990. Conference Proceedings., IEEE International Conference on*, 1990, pp. 14–19.

[27] P.S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Deep learning for monaural speech separation," in *Acoustics, Speech and Signal Processing, 2014 IEEE International Conference on*, 2014, pp. 1562–1566.

[28] S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher, "Block-coordinate frank-wolfe optimization for structural svms," *arXiv preprint arXiv:1207.4747*, 2012.

[29] B. McFee, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *14th annual Scientific Computing with Python conference*, July 2015, SciPy.

[30] S. Van Der Walt, J. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, pp. e453, 2014.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[32] A.C. Müller and S. Behnke, "Pystruct: learning structured prediction in python," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2055–2060, 2014.

[33] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 4, pp. 1462–1469, 2006.