DETECTING DROPS IN ELECTRONIC DANCE MUSIC: CONTENT BASED APPROACHES TO A SOCIALLY SIGNIFICANT MUSIC EVENT

Karthik Yadati, Martha Larson, Cynthia C. S. Liem, Alan Hanjalic
Delft University of Technology
{n.k.yadati,m.a.larson,c.c.s.liem,a.hanjalic}@tudelft.nl

ABSTRACT

Electronic dance music (EDM) is a popular genre of music. In this paper, we propose a method to automatically detect the characteristic event in an EDM recording that is referred to as a *drop*. Its importance is reflected in the number of users who leave comments in the general neighborhood of drop events in music on online audio distribution platforms like SoundCloud. The variability that characterizes realizations of drop events in EDM makes automatic drop detection challenging. We propose a two-stage approach to drop detection that first models the sound characteristics during drop events and then incorporates temporal structure by zeroing in on a watershed moment. We also explore the possibility of using the drop-related social comments on the SoundCloud platform as weak reference labels to improve drop detection. The method is evaluated using data from SoundCloud. Performance is measured as the overlap between tolerance windows centered around the hypothesized and the actual drop. Initial experimental results are promising, revealing the potential of the proposed method for combining content analysis and social activity to detect events in music recordings.

1. INTRODUCTION

Electronic dance music (EDM) is a popular genre of dance music which, as the name suggests, is created using electronic equipment and played in dance environments. Outside of clubs and dance festivals, EDM artists and listeners actively share music on online social platforms. Central to the enjoyment of EDM is a phenomenon referred to as "The Drop". Within the EDM community, a *drop* is described as a moment of emotional release, where people start to dance "like crazy" [12]. There is no precise recipe for creating a drop when composing EDM. Rather, a drop occurs after a *build*, a building up of tension, and is followed by the re-introduction of the full bassline [1]. A given EDM track may contain one or more drop moments. The designation "The Drop" is generally reserved for the overall phenomenon rather than specific drop events.

In this paper we address the challenge of automatically detecting a drop in a given EDM track. The social significance of the drop in the EDM context can be inferred, for instance, from the websites that compile a playlist of the best drops [1] . It is also evident from vivid social activity around drop events on online audio distribution platforms such as SoundCloud [2] . We also mention here a documentary, scheduled to be released in 2014, tracking the evolution of EDM as a cultural phenomenon, and titled *The Drop* [3] . Ultimately, the drop detection approach proposed in this paper could serve both EDM artists and listeners. For example, it would enable artists to compare drop creation techniques, and would also support listeners to better locate their favorite drop moments.

The challenge of drop detection arises from the high variability in different EDM tracks, which differ in their musical content and temporal development. Our drop detection approach uses audio content analysis and machine learning techniques to capture this variability. As an additional source of reference labels for classifier training, we explore the utility of drop-related social data in the form of *timed comments*, comments associated with specific time codes. We draw our data from SoundCloud, a music distribution platform that supports timed comments and is representative of online social sharing of EDM. The paper makes three contributions:

- We propose a two-step content-based drop detection approach.
- We verify the ability of the approach to detect *drops* in EDM tracks.
- We demonstrate utility of the social features (timed comments on SoundCloud) to reduce the amount of hand-labeled data needed to train our classifier.

The remainder of this paper is organized as follows. Section 2 discusses related work, and is followed by the presentation and evaluation of our method in sections 3 and 4. Section 5 provides a summary and an outlook towards future work.

[1] http://www.beatport.com/charts/top-10-edm-drops-feb1/252641

[2] http://soundcloud.com

[3] http://www.imdb.com/title/tt2301898/

2. RELATED WORK

Although Electronic Dance Music is a popular music genre attracting large audiences, it has received little attention in the music information retrieval research community. Research on EDM is limited to a small number of contributions. Here, we mention the most notable. Hockman et al. [5] propose a genre-specific beat tracking system that is designed to analyze music from the following EDM subgenres: Hardcore, Jungle, Drum and Bass. Kell et al. in [6] also apply audio content analysis to EDM in order to investigate track ordering and selection, which is usually carried out by human experts, i.e., Disc Jockeys (DJ). The work report findings on which content features influence the process of ordering and selection. A musicological perspective is offered by Collins in [3], who applies audio content analysis and machine learning techniques to empirically study the creative influence of earlier musical genres on the later ones using a date annotated database of EDM tracks, with specific focus on the sub-genres Detroit techno and Chicago house. Our work strives to redress the balance and give more attention to EDM. It draws attention to SoundCloud as an important source of music data and associated social annotations, and also to "The Drop", a music event of key significance for the audience of EDM.

The rise of social media has also seen the rise in availability of user-contributed metadata (e.g., comments and tags). Social tags have recently grown in importance in music information retrieval research. In [11], they were used to predict perceived or induced emotional responses to music. This work reports findings on the correlation between the emotion tags associated with songs on Last.fm—"happy", "sad", "angry" and "relax"—and the user emotion ratings for perceived and induced emotions. Social data is generally noisy, since generating precise labels is not users' primary motivation for tagging or commenting. However, this data can still prove useful as weak reference labels, reducing the burden of producing ground-truth labels for a large set of music tracks, which is an expensive and time consuming task. Social tags available on Last.fm have been used to automatically generating tags for songs [4]. An interesting direction of research is described in [13], where the authors use content-based analysis of the song to improve the tags provided by users. Existing work makes use of social tags that users assign to a song as a whole. In contrast, our work makes use of *timed comments* that users contribute associated with specific time points during a song.

Obtaining time-code level ground-truth labels for a large set of music tracks is an expensive and time consuming task. One way to obtain reference labels is to use crowdsourcing, where users are explicitly offered a task (e.g., label the type of emotion [9]). Our approach of using timed comments spares the expense of crowdsourcing. It has the additional advantage that users have contributed the comments spontaneously, i.e., they have not been asked to explicitly assign them, making them a more natural expression of user reactions during their listening experience.

3. PROPOSED APPROACH

Our proposed two-step approach is based on general properties of the "The Drop". As previously mentioned drops are characterized by a build up towards a climax followed by reintroduction of the bassline. We hypothesize that the switch will coincide with a structural segment that ends at a drop moment. For this reason, the first step in our approach is segmentation. However, not all segment boundaries are drops. For this reason, the second step in our approach is a content-based classification of segments that eliminates segments whose boundaries are not drop points. Figure 1 illustrates the two-stage approach, where we first segment to identify drop candidates and then classify in order to isolate candidates that are actually drop moments.

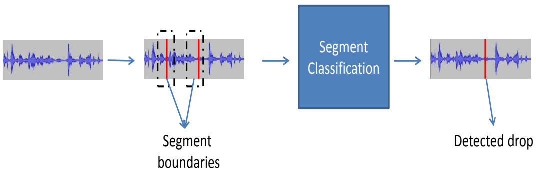


Figure 1. Two-stage approach to drop detection

The classification framework we propose to find drop events is illustrated in Figure 2. At the heart of the framework are the following modules: Segmentation, feature extraction, classification and evaluation.

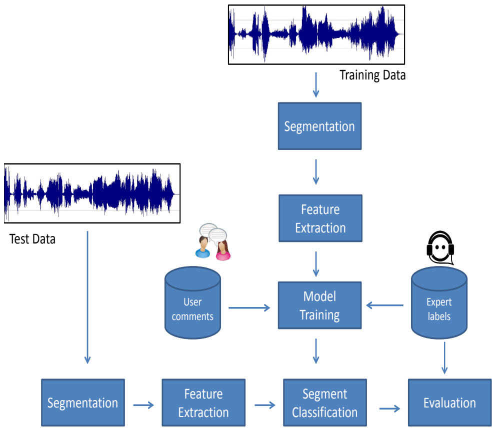


Figure 2. The proposed classification framework

3.1 Segmentation

The segmentation step carries out unsupervised segment boundary detection. Exploratory experiments revealed that the segmentation method proposed in [10] gives a good first approximation of the drops in an EDM track, and we have adopted it for our experiments. The method uses the chroma features computed from the audio track to identify the segment boundaries. We use the same parameters as used in [10]: 12 pitch classes, a window length of 209 ms, and a hop size of 139 ms. We carried out an intermediate evaluation to establish the quality of the drop candidates generated by the segmentation step alone. The average distance between the actual drop (ground-truth) and a segment boundary generated by our segmentation method is 2.5 seconds, and less than 8% of the drops are missed in our training set (described in Section 4.1).

3.2 Feature Extraction for Classification

An overview of the feature extraction process is illustrated in Figure 3.

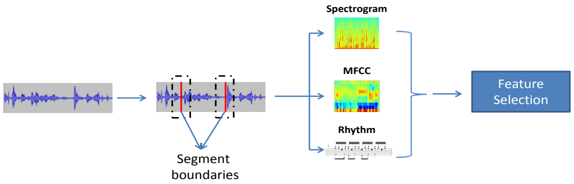


Figure 3. Feature extraction procedure

After segmentation, we extract content-based features from a fixed length window around the segment boundary. We use the following features: Spectrogram, MFCC and features related to rhythm. We adopt Mel-Frequency Cepstral Coefficients (MFCC) and features computed from the spectrogram because of their effectiveness. A unique feature of a drop is that it is preceded by a buildup or *build*. Figure 4 indicates that this buildup can be clearly observed in the spectrogram of an audio segment containing a drop. This provides additional motivation to use features computed from the spectrogram in our approach. We use the statistics computed from the spectrogram in our method (mean and standard deviations of the frequencies). For MFCC and spectrogram calculation, we use a window size of 50 msec with a 50% overlap with the subsequent windows. We use 13 coefficients for the MFCC. Due to a switch of rhythm at the drop moment, features related to rhythm are another important source of information. We use the rhythm related features: rhythm patterns, rhythm histogram, temporal rhythm histogram [8]. We concatenate the rhythm features, MFCC and statistics computed from the spectrogram into a single feature vector. Feature selection, following the approach of [2], is performed on the training data in order to reduce the dimensionality of the feature vector and also to ensure that we use the most informative features in the classification step.

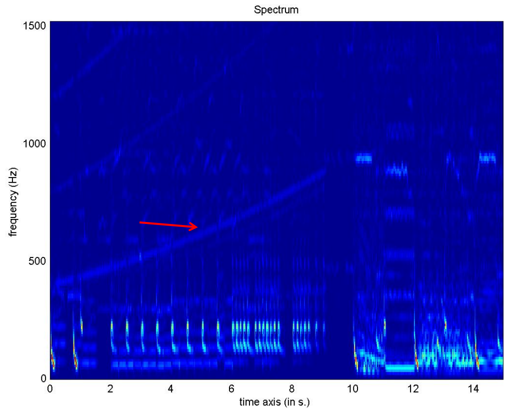


Figure 4. Spectrogram of an audio segment indicating a build (red arrow) towards a drop at 10 seconds.

3.3 Training and Classification

To train the classifier, we assign drop (1) vs. non-drop (0) labels to time-points in the track using two sources of information: high fidelity ground-truth (manual labels provided by an expert) and user comments (weak reference labels).

Prior to training the model, we map the ground-truth labels to the nearest segment boundaries. We note that the segmentation step reduces the search space for the drop, as we no longer search for it in the entire track, but focus on features around the segment boundaries. We use a binary SVM classifier with a linear kernel as our training algorithm.

3.4 Evaluation

Our method predicts time points in a track at which the drop occurs. We consider each detected drop to be a distinct drop. The fact that the drop can only be hypothesized at a segment boundary keeps detections from occurring close together, given that the average length of segments generated by our segmentation algorithm is 16.5 seconds.

In order to report the performance in terms of accuracy and precision, we utilize the F1-score. Although the drop is annotated as a point in the track, it is characterized by the music around the point. This aspect of the drop motivates our choice of using a tolerance window of varying temporal resolutions around the hypothesized drop and use temporal overlap to compute the F1-score. We follow these steps to compute the F1-score:

1. Place a tolerance window of size t seconds centered around the hypothesized (from our approach) and the reference drop (ground-truth).
2. Compute the number of true positives (tp), false positives (fp) and false negatives (fn) as illustrated in Figure 5 (the unit of measurement being seconds). Note that the numbers computed here are related to the number of seconds of overlap between the windows placed over the actual drop and the predicted drop. These are computed for every detected drop in the track.
3. Compute the F1-score using the following equation: $F1 = \frac{2tp}{2tp+fn+fp}$.
4. Repeat the above steps for different sizes of t. We use windows sizes of *t = 15 sec, 13 sec, 11 sec, 9 sec, 7 sec, 5 sec, 3 sec* to compute the F1 score.
5. If there is more than one drop in the track, repeat all the above steps and compute an average F1-score for each size of the window t.

4. EXPERIMENTS

We have proposed a classification framework for detecting drops in an EDM track. We use MIRToolbox [7] to extract features related to spectrogram and MFCC, while we

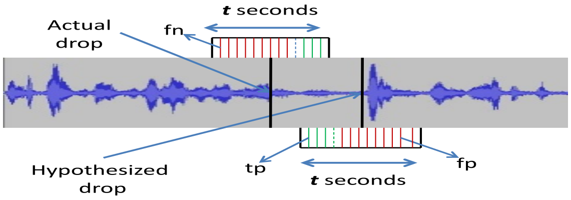


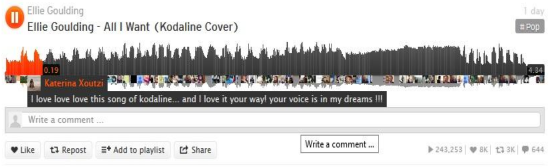
Figure 5. Illustration to compute true positive (tp), false positive (fp) and false negative (fn) using a rectangular window of size *t* seconds.

use the source code provided by the authors of [8] to extract features related to rhythm. We carry out feature selection with a mechanism, adopted from [2], that uses support vector machines to identify the most informative features. For the binary classification of drop vs. non-drop, we use a support vector machine classifier provided in LibSVM. The experiments have been designed to address the two research questions of this paper:

- Can our proposed approach detect drops successfully? (Section 4.3), and
- What is the utility of the timed comments in the limited presence of explicit ground-truth data? (Section 4.4)

4.1 Dataset

In order to evaluate our method, we collect music and social data from SoundCloud, which can be seen as a representative of modern online social audio distribution platforms. It allows users to upload, record and share their self-created music. One of the unique features of SoundCloud is that it allows users to comment at particular time-points in the sound. These comments are referred to as “timed comments”. Figure 6 illustrates a screenshot of the audio player on SoundCloud along with the timed comments.



I love love love this song of kodaline... and I love it your way! your voice is in my dreams !!!

Figure 6. Screenshot of the audio player on SoundCloud.

These comments offer a rich source of information as they are associated with a specific time-point and could indicate useful information about the sound difficult to infer from the signal. Table 1 illustrates a few example timed comments, which provide different kinds of information about the sound. These timed comments can be noisy with respect to their timestamps due to discrepancies between when users hear interesting events, and when they comment on them.

SoundCloud provides a well-documented API that can be used to build applications using SoundCloud data and information on select social features. In order to collect our dataset, we used the Python SDK to search for recent sounds belonging to the following three sub-genres of EDM: *dubstep*, *electro* and *house*. Using the returned list of track identification numbers, we download the track (if its allowed by the user who uploaded the sound) and the corresponding timed comments. We then filter out the comments which do not contain the word “drop”. At the end of the data collection process, we have a set of tracks belonging to the above mentioned genres, the associated timed comments containing the word “drop”, and the corresponding timestamp of the comment. Table 2 provides some statistics of the dataset.

Timestamp	Comment
01:21	Dunno what it is about this song, inspires me to make more tunes though! love it!
00:28	Love the rhythm!!
00:49	love that drop! nice bassline! nice vocals! epic!

Table 1. Examples of timed comments on SoundCloud.

Genre	# files	Aver. Duration	Aver. # comments	Aver. # drop comments	Aver. # drops
Dubstep	36	4 min.	278	4	3
Electro	36	3.6 min.	220	3	3
House	28	3.9 min.	250	5	2

Table 2. Statistics of the dataset

As we have filtered out the non-drop comments and all the tracks in the dataset have at least one drop comment, we can assume that there is at least one drop in each track. We use a dataset of 100 tracks with a split of 60–20–20 for the training, development and testing respectively.

4.2 Ground-truth annotations

As we are developing a learning framework to detect drops in an EDM track, we need reference labels for the time-points at which drops occur in our dataset, as mentioned previously. We utilize two sources of information: explicit ground-truth (high fidelity labels) and implicit ground-truth (user comments). In order to obtain high fidelity drop labels, one of the authors has listened to the 100 tracks and manually marked the drop points. The labeled points refer to the point where the buildup ends and the bassline is re-introduced. Instead of listening to the entire track, the author skips 30 seconds after he hears a drop as it is highly unlikely that a second drop would occur within 30 seconds. It took approximately 6 hours for the author to label the entire dataset. When computing F1-score in the experiments, we use the manual labels as ground-truth.

Explicit ground-truth labels are expensive as creating them requires experts to spend time and effort to listen to the tracks and mark the drop points. Relying on explicit ground-truth data also hampers the scalability of the dataset, as it would require much more time and effort from the annotators for a larger dataset. Keeping with the social nature of SoundCloud, users contribute comments, some which remark on the pretense or quality of a drop (Table 1). We investigate the possibility of using these timed comments as weak reference labels in predicting the drop. We refer to timed comments as *weak* reference labels owing to their noisy nature. For example, only 20 % of the drop comments in the training set are located at the actual drop in a track. Note that we treat each comment as a distinct drop. We have a total of 190 drops and 225 drop comments in our dataset. As we can see, there are more comments than the actual drops. Mapping multiple drop comments, which are nearer to each other, to a single time point is a consideration for the future.

4.3 Detecting drop using content-based features

In this experiment, we evaluate the performance of the content based features in detecting a drop using the explicit ground-truth labels. We compute the F1-score for each track separately. The F1-score is averaged if there is more than one drop in the track. In Table 3, we report three results: (1) F1-score, averaged across the entire dataset; (2) Highest F1-score for a single track and (3) Lowest F1-score for a single track. As mentioned before, we use windows of sizes *t = 3, 5, 7, 9, 11, 13, 15 sec*. The size of the window (t) represents the temporal precision to which the F1-score is reported. Observing the results for the average performance (first row of Table 3), we achieve a maximum F1 score of 0.71 for a 15 second tolerance window. However, we already achieve an F1 score greater than 0.6 for a tolerance window as small as 3 seconds. The second row of Table 3 illustrates the F1 scores for one single track which has the best drop detection and we observe that the F1 scores are high and go up to 0.96 for a 15 second tolerance window. The third row of Table 3 illustrates the F1 scores for one single track which has the worst drop detection and we observe that the F1 scores are very low, as it has more false positives. Moreover, the structure segment boundaries do not capture the drops particularly well in this track.

4.4 Utility of timed comments

Timed comments are an important source of information as they could indicate the time point where a drop occurs. Figure 7 illustrates a pipeline for the experiment to assess the utility of timed comments as weak reference labels. It is carried out in three stages labeled as (1), (2) and (3) in the figure. The stages are explained here. We divide the complete training set of N tracks into two mutually exclusive sets of n and $N - n$ tracks. Assuming that the n tracks have ground-truth labels, we train a model (1) and use it to classify the unlabeled segment boundaries from the $N - n$ tracks. We segment boundaries labeled positive by the classifier, which will be of low fidelity, and add them it to the training data. In the second stage (2), we use the expanded training data (n tracks + low fidelity positive segment boundaries) to predict the drop segments in the test set and compute the F1 score for evaluation. Then, the features computed from a window sampled around user drop comments are added to the training data. The data now includes features from the n tracks, and low fidelity predicted positive segment boundaries, and around sampled at user comments. We use this data to train a model (3) and use it to predict the drop segments in the test set and compute the F1 score for evaluation.

In this experiment, we use the following training data sizes which are expressed in terms of the number of tracks: $n = 5, 10, 20, 30, 40, 50$. F1 scores over different window sizes are computed to demonstrate the drop detection performance. Figure 8 illustrates the performance of the binary classifier when we have increasing sizes of training data. Due to space constraints, we illustrate the results only for one size of the tolerance window: 11 seconds. Difference in F1 scores when we add user comments is visualized in Figure 8. Inspecting the figure, we can say that reasonable F1 scores are obtained when we use $n = 30$ and $n = 40$ tracks as training set and a tolerance window size of 11 seconds. We observe that the F1 scores are lower than with explicit ground-truth annotations, which we attribute to the noise of user comments.

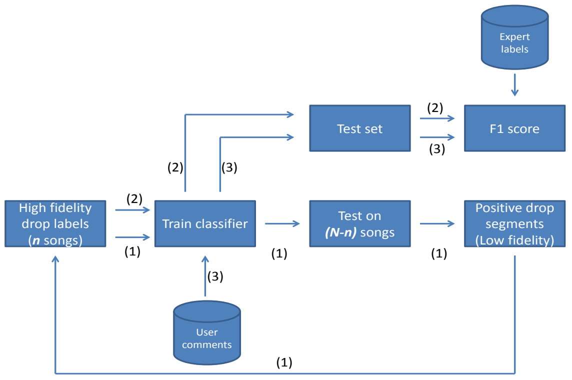


Figure 7. Procedure to assess the utility of timed comments in detecting drop.

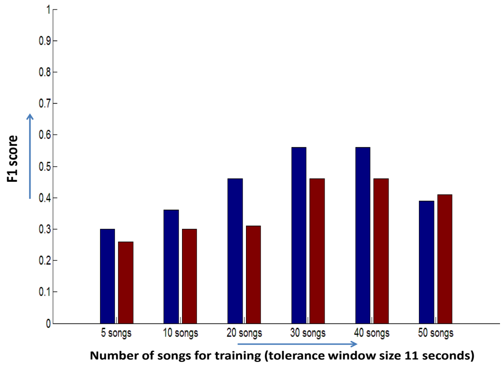


Figure 8. F1 scores for combining high fidelity ground-truth labels and user comments for a tolerance window size of 11 seconds and different training set sizes: 5 tracks, 10 tracks, 20 tracks, 30 tracks, 40 tracks, 50 tracks. First bar in each group indicates the results of stage (3) of the experiment and the second bar indicates the F1 score for the stage (2) of the experiment

5. CONCLUSION AND OUTLOOK

We have proposed and evaluated content-based approach that detects an important music event in EDM referred to as a drop. To this end, we have made use of music and user-contributed timed-comments from an online social audio

	3 sec	5 sec	7 sec	9 sec	11 sec	13 sec	15 sec
Average Performance	0.61	0.62	0.66	0.66	0.68	0.69	0.71
Track with Best Performanc	0.83	0.9	0.92	0.94	0.95	0.96	0.96
Track with Worst Performance	0.2	0.36	0.43	0.47	0.49	0.51	0.52

Table 3. Experimental results indicating the average, best and worst F1 scores for increasing window sizes

distribution platform: SoundCloud. We reported performance in terms of F1, using a tolerance window of varying time resolutions around the reference drop time-points and the drop time-points hypothesized by our approach. With a tolerance window of 5 seconds, which we estimate to be an acceptable size to listeners, we obtain an F1 score greater than 0.6. “Timed-comments”, contributed by users in association with specific time-codes were demonstrated to be useful as weak labels to supplement hand-labeled reference data. We achieved a reasonable accuracy using a standard set of music related features. One of the future steps would be to come up with a set of features which can model the variability and the temporal structure during drop events, which will in turn improve the accuracy. We concentrated on a subset of genres: dubstep, electro and house in this paper as these were the more popular genres on SoundCloud (in terms of number of comments). An immediate direction would be to expand the current dataset by including various sub-genres of EDM, e.g., techno and drum & bass.

Our work demonstrates that musical events in popular electronic music can be successfully analyzed with the help of time-level social comments contributed by users in online social sharing platforms. This approach to music event detection opens up new vistas for future research. Our next step is to carry out a user study with our drop detector aimed at discovering exactly how it can be of use to EDM artists and listeners. Such a study could also reveal the source of “noise” in the timed comments, allowing us to understand why users often comment about drops in neighborhoods far from where an actual drop has occurred. This information could in-turn allow us to identify the most useful drop comments to add to our training data. Further, we wish to widen our exploration of information sources that could possibly support drop detection to also include MIDI files that are posted by users online together with the audio. Currently, the availability of these files is limited, but we anticipate that they might be helpful for bootstrapping. Another source of information is a crowdsourcing, which could be used to identify drops directly, or to filter comments directly related to the drop, from less-closely related or unrelated comments.

6. ACKNOWLEDGEMENT

This research is supported by funding from the European Commission’s 7th Framework Program under grant agreement no. 610594 (CrowdRec) and 601166 (PHENICX).

7. REFERENCES

[1] M.J. Butler. *Unlocking the Groove: Rhythm, Meter, and Musical Design in Electronic Dance Music*. Profiles in popular music. Indiana University Press, 2006.

[2] Yi-Wei Chen and Chih-Jen Lin. Combining SVMs with various feature selection strategies. In *Feature Extraction*, volume 207 of *Studies in Fuzziness and Soft Computing*, pages 315–324. Springer Berlin Heidelberg, 2006.

[3] Nick Collins. Influence in early electronic dance music: An audio content analysis investigation. In *ISMIR*, pages 1–6, 2012.

[4] Douglas Eck, Paul Lamere, Thierry Bertin-Mahieux, and Stephen Green. Automatic generation of social tags for music recommendation. In *NIPS*, 2007.

[5] Jason Hockman, Matthew E. P. Davies, and Ichiro Fujinaga. One in the jungle: Downbeat detection in hardcore, jungle, and drum and bass. In *ISMIR*, pages 169–174, 2012.

[6] Thor Kell and George Tzanetakis. Empirical analysis of track selection and ordering in electronic dance music using audio feature extraction. In *ISMIR*, pages 505–510, 2013.

[7] Olivier Lartillot and Petri Toiviainen. Mir in matlab (ii): A toolbox for musical feature extraction from audio. In *ISMIR*, pages 127–130, 2007.

[8] Thomas Lidy and Andreas Rauber. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In *ISMIR*, pages 34–41, 2005.

[9] Erik M. Schmidt and Youngmoo E. Kim. Modeling musical emotion dynamics with conditional random fields. In *ISMIR*, pages 777–782, 2011.

[10] Joan Serrà, Meinard Müller, Peter Grosche, and Josep LLuis Arcos. Unsupervised music structure annotation by time series structure features and segment similarity. *IEEE Transactions on Multimedia*, PP(99):1–1, 2014.

[11] Yading Song, Simon Dixon, Marcus Pearce, and Andrea R. Halpern. Do online social tags predict perceived or induced emotional responses to music? In *ISMIR*, pages 89–94, 2013.

[12] John Steventon. *DJing for Dummies*. –For dummies. Wiley, 2007.

[13] Yi-Hsuan Yang, Dmitry Bogdanov, Perfecto Herrera, and Mohamed Sordo. Music retagging using label propagation and robust principal component analysis. In *WWW*, pages 869–876, 2012.