A Supervised Classification Approach for Note Tracking in Polyphonic Piano Transcription

Jose J. Valero-Mas, Emmanouil Benetos and José M. Iñesta [*]

Abstract

In the field of Automatic Music Transcription, note tracking systems constitute a key process in the overall success of the task as they compute the expected note-level abstraction out of a frame-based pitch activation representation. Despite its relevance, note tracking is most commonly performed using a set of hand-crafted rules adjusted in a manual fashion for the data at issue. In this regard the present work introduces an approach based on machine learning, and more precisely supervised classification, that aims at automatically inferring such policies for the case of piano music. The idea is to segment each pitch band of a frame-based pitch activation into single instances which are subsequently classified as active or non-active note events. Results using a comprehensive set of supervised classification strategies on the MAPS piano dataset report its competitiveness against other commonly considered strategies for note tracking as well as an improvement of more than +10 % in terms of F-measure when compared to the baseline considered for both frame-level and note-level evaluation.

Keywords: Note Tracking, Polyphonic Piano Transcription, Onset Detection, Supervised Classification, Machine Learning, Audio Analysis

1 Introduction

Automatic Music Transcription (AMT) stands for the process of automatically retrieving a high-level symbolic representation of the music content present in an audio signal (Grosche et al., 2012). This particular task has been largely studied and addressed by the Music Information Retrieval (MIR) field due to its considerable application in a number of tasks such as music preservation and annotation (Kroher et al., 2016), music similarity and retrieval (Lidy et al., 2010), and computational musicological analysis (Klapuri & Davy, 2007),

[*] This research work is partially supported by Universidad de Alicante through the FPU program (UAFPU2014–5883) and the Spanish Ministerio de Economía y Competitividad through project TIMuL (No. TIN2013–48152–C2–1–R, supported by EU FEDER funds). EB is supported by a UK RAEng Research Fellowship (grant no. RF/128). Jose J. Valero-Mas and José M. Iñesta are with the Pattern Recognition and Artificial Intelligence Group (PRAIg) of the Department of Sofware and Computer Systems, University of Alicante, Spain. Emmanouil Benetos is with the Centre for Digital Music, Queen Mary University of London, United Kindom. e-mail: jjvalero@dlsi.ua.es, emmanouil.benetos@qmul.ac.uk, inesta@dlsi.ua.es

among others. While the task of automatically transcribing monophonic music is largely considered to be solved, automatic transcription of polyphonic music still remains an open problem (Benetos et al., 2013).

With the sole exception of some particular systems as for instance the one by Berg-Kirkpatrick, Andreas, and Klein (2014), the majority of AMT systems comprise two stages (Benetos et al., 2013): an initial *multipitch estimation* (MPE) stage in which the system estimates the active pitches in each frame of the signal; and a *note tracking* (NT) stage that processes and refines the results of the former MPE step to obtain a higher-level description of the note events in terms of a discrete pitch value, onset, and offset. Thus, while the former stage aims at retrieving a *raw* pitch description of the signal, the latter acts as both a correction and segmentation stage for obtaining musically-meaningful representations (Cheng et al., 2015).

Multipitch estimation has been largely explored due to its relevance not only in AMT but also in fields such as source separation or score following (Duan et al., 2010). These systems retrieve a time-pitch representation typically referred to as *pitch activation* or *posteriorgram* that depicts the temporal evolution of the salience of each pitch band. In general, these techniques may be grouped in three different categories depending on the type of principle considered (Benetos et al., 2012): *(i)* feature-based methods that extract meaningful descriptors of the signal for later applying either heuristics or machine learning methods for the estimation; *(ii)* modelling the estimation within a statistical framework and thus address the problem as the estimation of the parameters of the distribution; and *(iii)* the spectrogram factorization paradigm that considers the initial time-frequency representation (generally, a spectrogram) as a matrix to be decomposed into a series of pitch templates and activations. The particular case of the latter approach has proved to be quite effective for MPE, and thus a number of strategies based on that principle have been proposed from which Non-negative Matrix Factorisation (NMF) and Probabilistic Latent Component Analysis (PLCA) stand out.

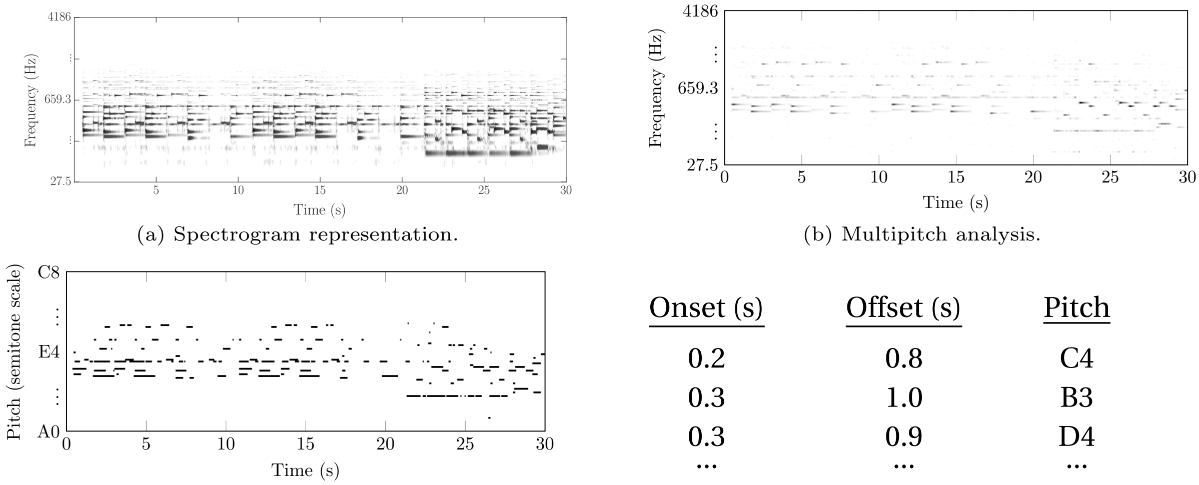
On the contrary, note tracking has not received that much attention despite its relevance in the overall success of the automatic music transcription task (Duan & Temperley, 2014). Note-level transcriptions are commonly obtained by binarizing the time-pitch representation and post-processing it with a set of minimum-length pruning processes for eliminating spurious detections and gap-filling stages for removing small gaps between consecutive pitches as, for instance, works by Benetos and Weyde (2015) or Iñesta and Pérez-Sancho (2013). Thus, our main criticism lies in the fact that note tracking strategies typically rely on hand-crafted rules. Hence, as opposed to such methods, in this work we consider and explore an approach based on Pattern Recognition and Machine Learning so that the system may automatically infer the proper strategy for performing the note tracking task.

More precisely, the present paper expands the initial work in Valero-Mas, Benetos, and Iñesta (2016) which explored, as a proof of concept, the use of supervised classification approaches for note tracking as a post-processing stage using as input a frame-level transcription. More precisely, a binary classifier was used to post-process an initial binarized posteriorgram by labelling the events as either active or non-active and thus obtain a note-level representation. The new contributions in this work with respect to the aforementioned proof-of-concept publication are: *(i)* the use of a larger amount of classification schemes for the testing the method; *(ii)* a comprehensive experimental set-up to assess the potential and capabilities of the proposed method; and *(iii)* a comparison with existing and published approaches commonly considered for the task.

The rest of the paper is structured as follows: Section 2 reviews related work on note tracking to contextualize our contribution; Section 3 presents the proposed method for note tracking; Section 4 addresses the experimental set-up and the evaluation methodology considered; Section 5 discusses the results obtained; finally, Section 6 concludes the work and introduces directions for future work.

2 Background on note tracking

The first step towards obtaining a note-level representation consists in binarizing the posteriogram estimation obtained with the multipitch analysis of the piece, i.e. the non-binary two-dimensional representation depicting the prominence of each pitch value of being present at a certain time stamp. This is typically done by applying a threshold to the pitch activations, i.e. the values over a certain threshold are considered active pitch elements while the ones below it are assumed to be silence. In some cases, the result of this binarization process is directly considered to be a high-level representation, namely frame-level transcription, as seen in works as Vincent, Bertin, and Badeau (2010) or Grindlay and Ellis (2011). Figure 1 shows a graphical example of the process.



(a) Spectrogram representation.

(b) Multipitch analysis.

(c) Binary frame-level representation obtained with a single thresholding stage.

Onset (s)	Offset (s)	Pitch
0.2	0.8	C4
0.3	1.0	B3
0.3	0.9	D4
...	...	...

(d) List of note events obtained with the note tracking process.

Figure 1: Example of a frame-level transcription using a simple thresholding stage applied to a multipitch analysis of an excerpt of piano music.

The use of such approach has the clear advantage of its conceptual simplicity. Nevertheless, they generally entail low performance results as they are not robust enough against errors that might occur in the MPE stage as, for instance, false positives or over-segmentation of long note events. In this regard, alternative techniques that post-process the initial binarization are also considered to address those types of errors. Most commonly, these techniques are based on combinations of *minimum-length pruning* processes for eliminating spurious detections and, occasionally, *gap-filling stages* for removing small gaps between consecutive note events. Quite often, these techniques are implemented as rule-based systems. For example, works by Dessein, Cont, and Lemaitre (2010) and Benetos and Weyde (2015) considered simple pruning stages for removing false detections, while the system in Bello, Daudet, and Sandler (2006) studied a more sophisticated set of rules comprising both pruning and gap-filling stages.

Probabilistic models have also been considered for the note tracking process. In this regard, Hidden Markov Models (HMMs) have reported remarkably good results in the literature: the work by Ryynänen and Klapuri (2005) considered HMMs to model note events in terms of their attack, sustain, and noise states; Cheng et al. (2015) also proposed a four-stage HMM to model the states of a musical note; finally, other works such as Poliner and Ellis (2007); Benetos and Dixon (2013); Cañadas-Quesada, Ruiz-Reyes, Vera-Candeas, Carabias-Orti, and Maldonado (2010) proposed systems in which binary pitch-wise HMM models are used for modelling events as either active or inactive.

Alternative methodologies to the commented ones may also be found in the literature. For instance, Raczyński, Ono, and Sagayama (2009) proposed a probabilistic model based on dynamic Bayesian networks which takes as input the result of an MPE analysis. Other examples are the work by Duan and Temperley (2014), which presented a system that models the NT issue as a maximum likelihood problem, or the one by Pertusa and Iñesta (2012) in which this task is addressed by favouring smooth transitions among partials using directed acyclic graphs. Finally, a last work to highlight due to its conceptual relation to the approach proposed in this paper is the one by Weninger, Kirst, Schuller, and Bungartz (2013). In that work a classification-based approach was presented in which a set of Support Vector Machines (SVMs) were trained on a set of low-level features obtained from the pitch activations obtained from a supervised NMF analysis for then performing the note tracking process.

It must be noted that, in general, MPE systems are rather imprecise in terms of timing. Examples of typical issues are their tendency to miss note onsets, mainly due to the irregularity of the signal during the attack stage, the over-segmentation of long notes or the merge of repeated notes (e.g., tremolo passages) into single events. Hence, the use of timing information in this context is clearly necessary and useful (Valero-Mas et al., 2017).

Under this premise some works have considered the use of onset information to address such issues. Examples of such works may be found in Marolt and Divjak (2002), which considered onset information for tackling the problem of tracking repeated notes, the work by Emiya, Badeau, and David (2008), in which onset information was used for segmenting the signal before the pitch estimation phase, the proposal by Iñesta and Pérez-Sancho (2013), which postprocessed the result of the MPE stage with the aim of correcting timing issues with onset information, or the system by Grosche et al. (2012), which also considered onset information under an HMM framework. Note that, while scarce, some works as the one by Benetos and Dixon (2011) have considered both onset and offset estimation systems for tackling these timing issues.

To the best of our knowledge, no previous work has considered the use of supervised classification as a note tracking approach in the context of music transcription. Thus, in this work we consider supervised classification for post-processing an initial note-level estimation to model and correct the note-level transcription errors committed. Conceptually, the idea is to derive a set of instances from the initial note-level estimation of an audio piece by temporally segmenting each pitch band using as delimiters the estimated onset events of the piece; each of these instances is represented by a set of features obtained from this initial note-level representation and the initial multipitch estimation; each of these instances is subsequently categorized as being active or inactive segments of notes, thus producing the post-processed note-level transcription. This proposed approach is thoroughly described in the following section.

3 Proposed method

Figure 2 shows the general workflow for the proposed system, with the area labelled as *Note tracking* being the one devoted to the proposed note tracking method. In this system, the audio signal to transcribe undergoes a series of concurrent processes: an MPE stage to retrieve the pitch-time posteriorgram $P(p, t)$, which is then binarized and post-processed to obtain a frame-level transcription $T_F\left(p, t\right)$ (binary representation depicting whether pitch p at time frame t is active), and an onset estimation stage that estimates a list of onset events $(o_n)_{n=1}^{L}$. These three pieces of information are provided to the proposed note tracking method which post-processes the initial frame-level transcription $T_F\left(p, t\right)$ using the onset events to retrieve the final note-level transcription $T_N(p, t)$. Note that this process is carried out in two different stages: *(i)* a first one that considers the onset events $(o_n)_{n=1}^{L}$ for segmenting frame-level representation $T_F\left(p, t\right)$ into a set of examples or instances (i.e., models of the objects to work with – in our case, these objects are the commented segments from the frame-level representation – and characterised by a collection of features or descriptors); and *(ii)* a second stage which classifies these instances as being active or inactive elements in the eventual note-level transcription $T_N(p, t)$.

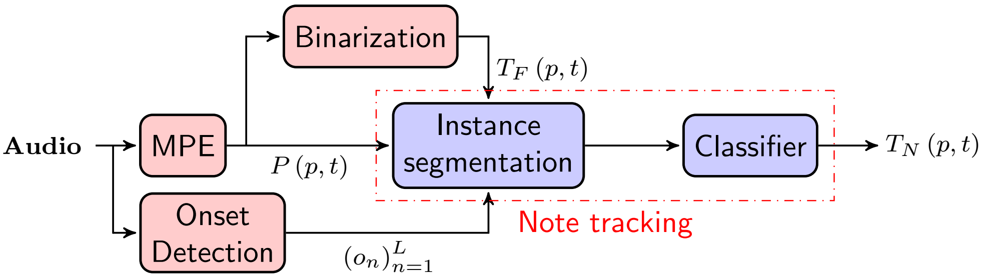


Figure 2: Set-up considered for the assessment of the classification-based note tracking method proposed.

It must be mentioned that, while the main contribution of this approach resides in how an initial frame-level transcription $T_F\left(p, t\right)$ is mapped into a set of instances to be classified, we present all system sub-components of Fig. 2 in the following sections as they constitute our entire note tracking workflow.

3.1 Multipitch Estimation

The first step of note tracking proposal is the multipitch analysis of the audio music piece to retrieve the pitch-time posteriorgram $P(p, t)$, for which we consider the system by Benetos and Weyde (2015). This system belongs to the Probabilistic Latent Component Analysis (PLCA) family of MPE methods and ranked first in the 2015 evaluations of the MIREX Multiple-F0 Estimation and Note Tracking Task[1]. PLCA is a spectrogram factorization variant which considers a normalised input spectrogram $V_{\omega,t}$ as a bivariate probability distribution $P(\omega, t)$ (here, ω stands for log-frequency and t for time). PLCA subsequently decomposes the bivariate distribution into a series of basis spectra and component activations. In the context of MPE, the component activations correspond to a probability of having an active pitch at a given time frame, and the basis to the spectrum of each pitch.

This particular system takes as input representation a variable-Q transform (VQT) and decomposes into a series of pre-extracted log-spectral templates per pitch, instrument source, and tuning deviation from ideal tuning. Outputs of the model include a pitch activation probability $P(p, t)$ (p stands for pitch in MIDI scale), as well as distributions for instrument contributions per pitch and a tuning distribution per pitch over time. The unknown model parameters are iteratively estimated using the Expectation-Maximization (EM) algorithm (Dempster et al., 1977), using 30 iterations in this implementation. For this particular study we consider a temporal resolution of 10 ms for the input time-frequency representation and output pitch activation and $|\mathcal{P}| = 88$ pitch values.

Finally, in order to retrieve the frame-level transcription $T_F(p, t)$, the pitch-time posteriorgram $P(p, t)$ obtained is processed as follows: first of all, $P(p, t)$ is normalized to its global maximum so that $P(p, t) \in [0, 1]$; then, for each pitch value $p_i \in p$, a median filter of 70 ms of duration is applied over time to smooth the detection; after that, the resulting posteriorgram is binarized using a global threshold value of $\theta = 0.1$ as it is the value which maximizes the note tracking figure of merit (to be introduced and commented in Section 4.2) after binarization; finally, a minimum-length pruning filter of 50 ms is applied to remove spurious detected notes.

3.2 Onset Detection

This process is devoted to obtaining the start times of note events present in the music signal at issue by means of an onset detection algorithm. For that we select four representative methods found in the literature for the detection of such events: a simple Spectral Difference (SD), the Semitone Filter-Bank (SFB) method by Pertusa, Klapuri, and Iñesta (2005), the SuperFlux (SF) algorithm by Böck and Widmer (2013b, 2013a), and Complex Domain Deviation (CDD) by Duxbury, Bello, Davies, and Sandler (2003). All these processes retrieve a list $(o_i)_{i=1}^{L}$ whose elements represent the time positions of the L onsets detected in the signal.

SD tracks changes in the spectral content of the signal by obtaining the difference between consecutive values of the magnitude spectrogram. Increases in such measure points out the presence of onset information in the frame under analysis.

SFB applies a harmonic semitone filter bank to each analysis window of the magnitude spectrogram and retrieves the energy of each band (root mean square value); a first-order derivative is then applied to each band; negative results are filtered out as only energy increases may point out onset information; finally, all bands are summed to obtain a function

[1] http://www.music-ir.org/mirex/wiki/MIREX_HOME

whose peaks represent the onset events.

SF expands the idea of the spectral flux signal descriptor by substituting the difference between consecutive analysis windows by tracking spectral trajectories in the spectrum together with a morphological dilation filtering process. This suppresses vibrato articulations in the signal which tend to increase false positives.

CDD combines the use of magnitude and phase information of the spectrum for the estimation. Basically, this approach aims at predicting the value of the complex spectrum (magnitude and phase) at a certain temporal point by using the information from previous frames; the deviation between the predicted and the actual spectrum values points out the possible presence of onsets.

3.3 Segmentation

As mentioned, the proposed note tracking strategy requires three sources of information, which are retrieved from the additional processes explained in the previous sections: the pitch-time posteriorgram $P(p,t)$, where p and t correspond to the pitch and time indices respectively, retrieved from the MPE stage; a frame-level transcription $T_F(p,t)$ obtained from the binarization and basic post-processing of $P(p,t)$; and an L-length list $(o_n)_{n=1}^{L}$ of the estimated onset events in the piece. Additionally, let $T_R(p,t)$ be the ground-truth piano-roll representation of the pitch-time activations of the piece, which is required for obtaining the labelled examples of the training set.

The initial binary frame-level transcription $T_F(p,t)$ can be considered a set of $|\mathcal{P}|$ binary sequences of $|t|$ symbols, where $|\mathcal{P}|$ and $|t|$ stand for the total number of pitches and frames in the sequence respectively. In that sense, we may use the elements $(o_n)_{n=1}^{L}$ as delimiters for segmenting each pitch band $p_i \in \mathcal{P}$ in $L+1$ subsequences. This process allows to segment frame-level transcription $T_F(p,t)$ with the onset information and express it as follows:

$$T_F(p_i,t) = T_F(p_i, 0:o_1) \;||\; T_F(p_i, o_1:o_2) \;||\; ... \;||\; T_F(p_i, o_L:|t|-1) \tag{1}$$

where $||$ represents the concatenation operator.

Each of these onset-based $L+1$ subsequences per pitch are further segmented to create the instances for the classifier. The delimiters for these segments are the points in which there is a change in the state of the binary sequence, i.e. when there is a change from 0 to 1 (inactive to active) or from 1 to 0 (active to inactive). Mathematically, for the onset-based subsequence $T_F(p_i, o_n : o_{n+1})$ the $|C|$ state changes are obtained as:

$$C = \{t_m : T_F(p_i, t_m) \neq T_F(p_i, t_{m+1})\}_{t_m=o_n}^{o_{n+1}} . \tag{2}$$

Thus, the resulting $|C|+1$ segments, which constitute the instances for the classifier, may be formally enunciated as:

$$T_F(p_i, o_n : o_{n+1}) = T_F(p_i, o_n : C_1) \;||\; ... \;||\; T_F(p_i, C_{|C|} : o_{n+1}) . \tag{3}$$

Figure 3 illustrates graphically this procedure. In this example, for frame-level transcription $T_F(p,t)$, in the interval given by $[o_n, o_{n+1}]$ and pitch p_i, there are $|C| = 4$ state changes (i.e., changes from active to inactive or viceversa). Hence we obtain $|C|+1 = 5$ subsequences.

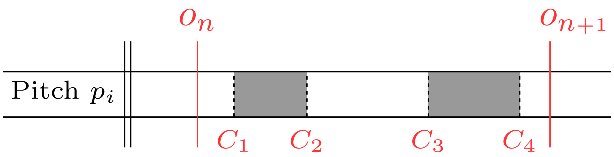


Figure 3: Conceptual example of the segmentation of the onset-based subsequence $T_F(p_i, o_n : o_{n+1})$ into instances. The grey and white segments depict sequences of 1 and 0, respectively.

So far we have performed the segmentation process based only on the information given by $T_F(p,t)$. Thus, at this point we are able to derive a set of instances that may serve as test set since they are not tagged according to the ground-truth piano roll $T_R(p,t)$. However, in order to produce a training set using the labels in $T_R(p,t)$, an additional step must be performed. For that we *merge* the pieces of information from both $T_F(p,t)$ and $T_R(p,t)$ representations, which we perform by obtaining the C set of delimiters as:

$$C = C_{T_F} \cup \{t_m : T_R(p_i, t_m) \neq T_R(p_i, t_{m+1})\}_{t_m=o_n}^{o_{n+1}} \tag{4}$$

where C_{T_F} represents the segmentation points obtained from $T_F(p,t)$. This need for merging these pieces of information in shown in Fig. 4: if we only took into consideration the breakpoints in $T_F(p_i,t)$ (i.e., the band labelled as *Detected*), subsequence $T_F(p_i, t_a : t_b)$ would have two labels if checking the figure labelled as *Annotation* – subsequence $T_F(p_i, t_a : t_c)$ should be labelled as non-active and $T_F(p_i, t_c : t_b)$ as active. Thus, we require these additional breakpoints to further segment the subsequences and align them with the ground-truth labels to produce the training set. Again, note that this process is not required for the test set since evaluation is eventually done in terms of note tracking performance and not as classification accuracy.

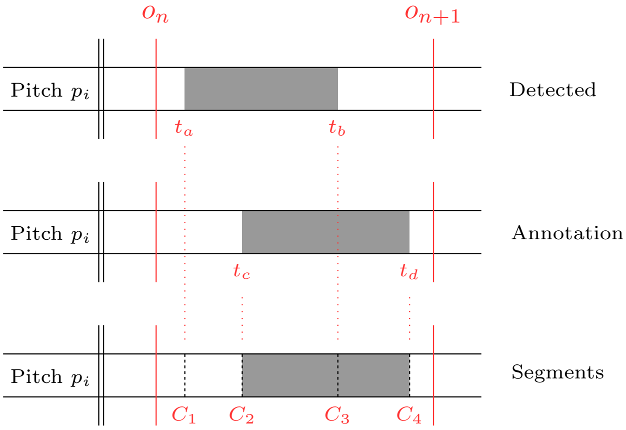


Figure 4: Conceptual example of the segmentation and labelling process for the training corpus. Breakpoints t_a and t_b from frame-level transcription $T_F(p_i,t)$ – labelled as *Detected* – together with breakpoints t_c and t_d from ground-truth piano roll $T_R(p_i,t)$ – labelled as *Annotation* – are considered for segmenting sequence $p_i \in \mathcal{P}$. Labels are retrieved directly from $T_R(p,t)$. For each case, grey and white areas depict sequences of 1 and 0, respectively.

Once the segmentation process has been performed, a set of characteristics is extracted for each of the instances. This set comprises features directly derived from the *geometry* of the instance (i.e., absolute duration or duration relative to the inter-onset interval), others derived from the frame-level transcription $T_F(p,t)$, as its distance to previous and posterior onsets, and others related to the posteriorgram $P(p,t)$ as the average energy in both the current and octave-related bands. No pitch information is included as feature, thus classification is performed independently of the pitch value. We assume that these features (both temporal and pitch salience-based descriptors) are able to capture relevant characteristics of the note tracking process. Table 1 describes the features considered and Fig. 5 graphically shows their obtaining process.

Table 1: Summary of the features considered. Operator $\langle\cdot\rangle$ retrieves the average value of the elements considered.

Feature	**Definition**	**Description**
Δt	$C_{m+1} - C_m$	Duration of the block
Δo_n	$C_m - o_n$	Distance between previous onset and the starting point of the block
Δo_{n+1}	$o_{n+1} - C_{m+1}$	Distance between end of the block and the posterior onset
$\mathcal{D}$	$\frac{\Delta t}{o_{n+1}-o_n}$	Occupation ratio of the block in the inter-onset interval
E	$\langle P(p_i, C_m : C_{m+1})\rangle$	Mean energy of the multipitch estimation in current band
E_l	$\langle P(p_i - 12, C_m : C_{m+1})\rangle$	Mean energy of the multipitch estimation in previous octave
E_h	$\langle P(p_i + 12, C_m : C_{m+1})\rangle$	Mean energy of the multipitch estimation in next octave

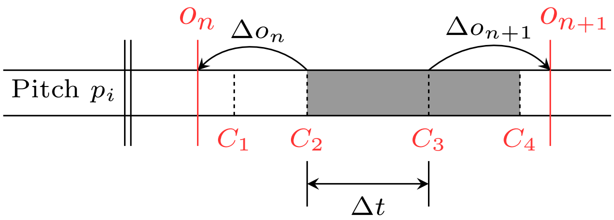


Figure 5: Graphical representation of the descriptors considered. In this conceptual example, the instance being characterized is $T_F(p_i, C_2 : C_3)$.

To avoid that the considered features may span for different ranges, we normalize them: energy descriptors (E, E_l, and E_h) are already constrained to the range $[0,1]$ as the input posteriorgram is normalised to its global maximum (cf. Section 4 in which the experimentation is described); occupation ratio $\mathcal{D}$ is also inherently normalized as it already represents a ratio between two magnitudes; absolute duration Δt and distance features Δo_n and Δo_{n+1} are manually normalised using the total duration of the sequence as a reference.

Finally, in an attempt to incorporate *temporal knowledge* in the classifier, we include as additional features the descriptors of the instances surrounding the one at issue (previous and/or posterior ones). To exemplify this, let us take the case in Fig. 5. Also consider to include a temporal context that of one previous and one posterior windows to the instance to be defined. To do so, and for the precise case of instance $T_F(p_i, C_2 : C_3)$, we take into account the features of both neighbouring instances $T_F(p_i, C_1 : C_2)$ and $T_F(p_i, C_3 : C_4)$.

3.4 Classifier

The proposed approach models the note tracking problem as a binary classification task in which the instances must be tagged as being either active or inactive (i.e., pitch activations in the audio signal). For that, the classifier requires both the set of instances to be classified and the reference data to create the classification model (i.e., the test and train data, respectively) derived from the process in Section 3.3 while retrieves the corresponding label (active/inactive) for each test instance.

As the note tracking strategy is not designed for any particular classification model, we now list the different classifiers we experimented with and whose performance will be later assessed and compared in Section 5. Note that, while the considered classification strategies are now introduced, the reader is referred to works by Bishop (2006) and Duda, Hart, and Stork (2001) for a thorough description of the methods:

1. Nearest Neighbour (NN): Classifier based on dissimilarity which, given a labelled set of samples $\mathcal{T}$, assigns to a query x' the class of sample $x \in \mathcal{T}$ that minimizes a dissimilarity measure $d(x, x')$. Generalising, if considering k neighbours for the classification (kNN rule), x' is assigned the mode of the individual labels of the k nearest neighbours.

2. Decision Tree (DT): Classifier that performs separation of the classes by iteratively partitioning the search space with simple decisions over the features in an individual fashion. The resulting model may be represented as a tree in which the nodes represent the individual decisions to be evaluated and the nodes contain the classes to assign.

3. AdaBoost (AB): Ensemble classifier based on the linear combination of weak classification schemes. Each weak classifier is trained on different versions of the training set $\mathcal{T}$ that basically differ on the weights (classification relevance or importance) given to the individual instances.

4. Random Forest (RaF): Ensemble-based classification scheme that categorizes query x' considering the decisions of one-level decision trees (decision stumps) trained over the same training set $\mathcal{T}$. The class predicted by the ensemble is the mode of the individual decisions by the stumps.

5. Support Vector Machine (SVM): Classifier that seeks a hyperplane that maximizes the margin between the hyperplane itself and the nearest samples of each class (support vectors) of training set T. For non-linearly separable problems, this classifier relies on the use of Kernel functions (i.e., mapping the data to higher-dimensional spaces) to improve the separability of the classes.

6. Multilayer Perceptron (MLP): Particular topology of an artificial neural network parametric classifier. This topology implements a feed-forward network in which each neuron in a given layer is fully-connected to all neurons of the following layer.

4 Experimentation

This part of the work introduces the experimentation carried out to assess the performance of our proposed note tracking method and its comparison with other existing methods. For that, we initially introduce the corpora considered, then we explain the figures of merit typically used for assessing note tracking systems, after that we introduce the parameters considered for our note tracking approach, and finally we list and explain other alternative note tracking strategies from the literature for the comparison of the results obtained.

4.1 Datasets

In terms of data, we employ the MAPS database (Emiya et al., 2010), which comprises several sets of audio piano performances (isolated notes, chords, and complete music works) synchronised with their MIDI annotations. For comparative purposes we reproduce the evaluation configuration in Sigtia, Benetos, and Dixon (2016). In that work the assessment was restricted to the subset of the MAPS collection of complete music works. This subset comprises 270 music pieces, out of which 60 were directly recorded using a Yamaha Disklavier piano under different recording conditions (these pianos are able to export both the audio recording and the ground-truth MIDI file) and the rest were synthesized from MIDI emulating different types of piano sounds. Within their evaluation, the data was organized considering a 4-fold cross validation, with 216 out of the 270 music pieces used for training and 54 music pieces for testing. The precise description of the folds can be found in `http://www.eecs.qmul.ac.uk/~sss31/TASLP/info.html`. Additionally, only the first 30 seconds of each of the files are considered for the experimentation as done in other AMT works, which gives up to a corpus with a total number of 72,585 note events. Table 2 summarizes the number of note events for each train/test fold.

Table 2: Description in terms of the number of note events for each train/test partition of the different folds considered.

	Fold 1	**Fold 2**	**Fold 3**	**Fold 4**
Train	59,563	59,956	54,589	60,527
Test	13,022	12,629	17,996	12,058

4.2 Evaluation metrics

For the evaluation of the proposed method we consider the methodology described in the *Multiple-F0 Estimation and Note Tracking* task which is part of the *Music Information Re-*

trieval Evaluation eXchange (MIREX) public evaluation contest[2]. The general idea behind this methodology is to assess how similar the obtained transcription $T_N(p,t)$ is to its corresponding ground-truth piano-roll representation $T_R(p,t)$. Additionally, as the proposed note tracking strategy considers the use of onset information, we also assess the performance of the onset detectors to evaluate the correlation between the goodness of the onset estimation algorithm and the note tracking task.

Regarding onset information, an estimated event is considered to be correct if its corresponding ground-truth annotation is within a ±50 *ms* window of it Bello et al. (2005).

For the case of note tracking evaluation we consider two evaluation methodologies to compare transcription $T_N(p,t)$ against ground-truth piano-roll $T_R(p,t)$: *(i)* a *frame-based* assessment that evaluates the correctness of the estimation by comparing both representations in a frame-by-frame basis; and *(ii)* a *note-based* evaluation that assesses the performance of the system by comparing the two representations in terms of note events defined by an onset, an offset, and a discrete pitch value. While the latter metric is the proper one to evaluate a note tracking approach as the one proposed, the former assessment can also provide valuable information to understand the performance of the method.

For the *frame-based* evaluation, a pitch frame estimated as active is considered to be correct if it matches an active pitch annotation within less than half semitone (±3% in terms of pitch value), considering a temporal resolution of 10 *ms*. As of *note-based* evaluation, we restrict ourselves to the onset-only note-based figure of merit as we are not considering note offsets; in our case, a detected note event is assumed to be correct if its pitch matches the corresponding ground-truth pitch and its onset is within ±50 *ms* of the corresponding ground-truth onset (Bay et al., 2009).

Based on the above criteria, and following the evaluation strategy of (Bay et al., 2009), we use the F-measure (F_1) as the main figure of merit, which properly summarises the overall performance of the method (in terms of correct, missed, and overestimated events) into one single value:

$$\mathrm{F}_1 = 2 \cdot \frac{\mathrm{N}_{\mathrm{OK}}}{\mathrm{N}_{\mathrm{DET}} + \mathrm{N}_{\mathrm{GT}}} \quad , \tag{5}$$

where N_{OK} stands for the number of correctly detected events (frames, onsets or notes, depending on the case), $\mathrm{N}_{\mathrm{DET}}$ for the number of total events detected, and N_{GT} the total amount of ground-truth events. This metric is obtained for each single recording to then obtain the general performance by averaging across recordings in each fold.

4.3 Parameters considered

This section introduces the analysis parameters of the different onset estimation methods and classifiers considered for the note tracking approach proposed in Section 3.

4.3.1 Onset estimation

The analysis parameters of the different onset estimation algorithms are set to the default values in their respective implementations: SFB considers windows of 92.8 *ms* with a tem-

[2] http://www.music-ir.org/mirex/wiki/MIREX_HOME

poral resolution of 46.4 ms; SF considers smaller windows of 46.4 ms with a higher temporal granularity of 5.8 ms; SD and CDD both consider windows of 11.6 ms with also a temporal resolution of 5 ms. Additionally, as all of them comprise a final thresholding stage, we test 25 different values equally spaced in the range $(0, 1)$ to check the influence of that parameter. From this analysis we select the value that maximizes the onset estimation in the data collection at issue for then use it in the note tracking stage. Finally, the onset lists $(o_n)_{n=1}^{L}$ are processed with a merging filter of 30 ms of duration to avoid overestimation issues as in Böck, Krebs, and Schedl (2012). Such filtering process takes all onset events that fall within a temporal lapse of 30 ms and retrieves a single onset event that represents its average value.

We also consider two additional situations regarding the origin of the onset information: a first one in which we consider ground-truth onset events and a second one in which the onset description is obtained by sampling a random distribution. Considering these three situations (i.e., automatic estimation, ground-truth events, and random distribution) allows us to assess the potential improvement that may be achieved with the proposed note tracking approach when considering the most accurate onset information (the ground-truth one) and compare it to the results achieved with the estimated events or the random onset description.

4.3.2 Classifiers

We now introduce the precise configuration of the classifiers considered for our note tracking approach:

1. Nearest Neighbour (NN): We restrict to one single nearest neighbour (i.e., 1NN) and consider the Euclidean distance as dissimilarity measure.

2. Decision Tree (DT): We consider the Gini impurity as the measure to perform the splits in the tree and set one sample per leaf (i.e., when a leaf contains more than one example, it becomes a node of the tree).

3. AdaBoost (AB): The weak classifiers considered for this ensemble-based scheme are decision trees.

4. Random Forest (RaF): For our experiments with this algorithm, the number of decision stumps is fixed to 10.

5. Support Vector Machine (SVM): We consider a radial basis function (RBF) for the kernel function, which constitutes a typical approach with this classifier.

6. Multilayer Perceptron (MLP): The configuration considered for this case is a single-layer network comprising 100 neurons with rectified linear unit (ReLU) activations and a softmax layer for the eventual prediction.

Note that the interest of the work lies in the exploration of the classification-based scheme rather than in parameter optimization. In that sense, the algorithms considered are directly taken from the Scikit-learn Machine Learning library (Pedregosa et al., 2011).

4.4 Comparative approaches

In order to perform comparative experiments with other note tracking methods from the literature, we also consider the use of pitch-wise two-state Hidden Markov Models (HMMs) as in Poliner and Ellis (2007). HMMs constitute a particular example of statistical model in which it is assumed that the system at issue can be described as a Markov process (i.e., a model for which the value of a given state is directly influenced by the previous one) with a set of unobservable states. In this work we replicate the scheme studied in the aforementioned work: we define a set of 88 HMMs (one per pitch considered) with two hidden states, active or inactive step; each HMM is trained by counting the type of transition between consecutive analysis frames (i.e., all combinations of transitioning from an active/inactive frame to an active/inactive one) of the elements of the training set; decoding is then performed on the test set using the Viterbi algorithm (Viterbi, 1967).

Finally, we also compare the proposed method with the results obtained by Sigtia et al. (2016) as we both replicate their experimental configuration and consider the same PLCA-based MPE method (cf. Section 3.1 for the description of the method). This consideration is mainly motivated by the fact that the aforementioned work constitutes a very recent method that tackles note-level transcription by implementing a polyphonic Music Language Model (MLM) based on a hybrid architecture of Recurrent Neural Networks (RNNs, a particular case of neural networks that model time dependencies) and a Neural Autoregressive Distribution Estimation (NADE, a distribution estimator for high dimensional binary data).

5 Results

This section presents the results obtained with the proposed experimental scheme for both the onset detection and note tracking methods, organized in two different subsections for facilitating their comprehension. The figures shown in each of them depict the average value of the considered figure of merit obtained for each of the cross-validation folds.

5.1 Onset detection

Firstly, we study the performance of the different onset detection methods considered. The aim is to assess the behaviour of these algorithms on the data considered to later compare the performance of the proposed note tracking method when considering different onset descriptions of the signal. For the assessment of the onset detectors we only consider the training set (we assume that test partition is not accessible at this point) and we assume that the conclusions derived from this study are applicable to the test set as they represent the same data distribution. In these terms, Fig. 6 graphically shows the average F_1 of the folds considered by the different onset estimation algorithms used as the selection threshold θ varies.

An initial remark to point out is the clear influence of the threshold parameter of the selection stage in the performance of the onset estimation methods. In these terms, SFB arises as the one whose performance is more affected by this selection stage, retrieving performance values that span from a completely erroneous estimation of $\mathrm{F}_1 \approx 0$ to fairly

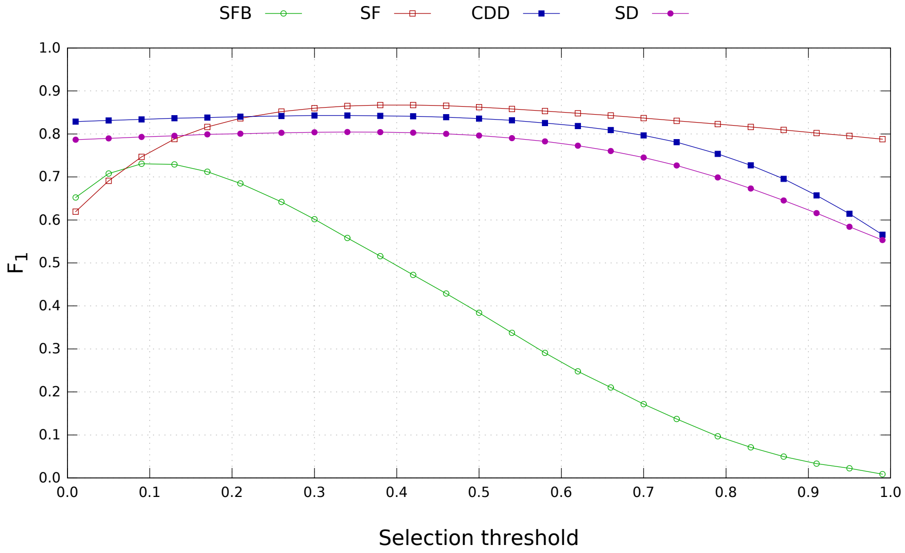


Figure 6: Onset detection results in terms of F_1 when varying the selection threshold. Acronyms in the legend stand for each onset estimation method: SFB for Semitone Filter-Bank, SuF for SuperFlux, CDD for Complex Domain Deviation, and SD for Spectral Difference.

accurate results of $F_1 \approx 0.75$. Attending to its performance, we select a threshold of $\theta = 0.13$ that reaches an approximate value of $F_1 = 0.75$ in the onset detection task.

SD and CDD show a totally opposite behaviour to the SFB method: these algorithms show a relatively steady performance for the threshold values studied with goodness figures of $F_1 \approx 0.8$ that only decrease to a performance of $F_1 \approx 0.5$ when the selected threshold approaches the unit. It can be seen that the CDD method shows a slightly better performance than the SD one, possibly due to the use of phase information for the estimation. For these two methods we find the local maxima when selecting threshold values of $\theta = 0.34$ of the SD methods and $\theta = 0.30$ for the CDD one, retrieving performances of $F_1 \approx 0.80$ and $F_1 \approx 0.82$ for the SD and CDD algorithms, respectively.

Finally, the SuF method also presents a very steady performance for all threshold values studied with the particular difference that the performance of the onset estimation degrades as the threshold value considered for the selection is reduced. Also, it must be pointed out that this algorithm shows the best performance among all studied methods when the selection stage is properly configured. In this case we select $\theta = 0.38$ as the threshold value that maximizes the performance of the algorithm.

5.2 Note tracking

Having analysed the performance of the considered onset selection methods, we now focus on the proposed note tracking approach. Table 3 shows the average results obtained (both *frame-based* and *note-based* assessments) with the cross-validation scheme considered for the proposed note tracking method with different classification strategies and numbers of adjacent instances. Note that the different onset detection methods used the thresholds that optimize their respective performance (i.e., the ones previously commented). These onset estimators are denoted with the same acronyms as above while the particular case when considering ground-truth onset information is denoted as GT.

Table 3: Average note tracking results in terms of F_1 for the proposed method. Notation (x, y) represents the number of previous and posterior additional instances considered. Bold figures highlight the best performing configuration per onset estimator and number of surrounding windows considered.

		GT		**SD**		**SFB**		**SuF**		**CDD**		**Random**	
		Frame	Note	Frame	Note	Frame	Note	Frame	Note	Frame	Note	Frame	Note
$(0, 0)$	NN	0.64	0.63	0.61	0.56	0.60	0.57	0.64	0.62	0.62	0.56	0.58	0.47
	DT	0.60	0.56	0.58	0.50	0.57	0.51	0.60	0.55	0.59	0.51	0.53	0.37
	RaF	0.66	0.65	0.63	0.57	0.62	0.58	0.66	0.64	0.63	0.58	0.57	0.48
	AB	0.56	0.60	0.53	0.52	0.51	0.54	0.56	0.59	0.54	0.53	0.46	0.45
	SVM	0.60	0.67	0.58	**0.61**	0.57	**0.62**	0.60	0.66	0.57	**0.62**	0.57	**0.61**
	MLP	**0.67**	**0.69**	**0.65**	0.60	**0.63**	0.61	**0.67**	**0.68**	**0.66**	0.61	**0.61**	0.57
$(1, 1)$	NN	0.65	0.69	0.61	0.56	0.60	0.59	0.63	0.62	0.61	0.57	0.57	0.47
	DT	0.62	0.59	0.60	0.50	0.59	0.52	0.62	0.54	0.60	0.50	0.53	0.37
	RaF	0.68	0.70	0.64	0.58	0.63	0.60	0.66	0.64	0.64	0.59	0.58	0.50
	AB	0.57	0.61	0.55	0.56	0.52	0.56	0.56	0.59	0.55	0.56	0.48	0.47
	SVM	0.58	0.69	0.56	0.58	0.54	**0.64**	0.57	0.64	0.56	0.58	0.56	**0.58**
	MLP	**0.70**	**0.72**	**0.66**	**0.60**	**0.65**	0.62	**0.68**	**0.66**	**0.66**	**0.61**	**0.60**	0.54
$(2, 2)$	NN	0.65	0.70	0.60	0.57	0.59	0.58	0.63	0.63	0.61	0.57	0.57	0.46
	DT	0.62	0.59	0.59	0.49	0.59	0.51	0.61	0.53	0.60	0.50	0.53	0.37
	RaF	0.68	0.70	0.63	0.58	0.63	0.59	0.66	0.64	0.64	0.58	0.57	0.50
	AB	0.59	0.63	0.55	0.55	0.53	0.57	0.57	0.59	**0.66**	0.56	**0.59**	0.44
	SVM	0.57	0.70	0.60	**0.62**	0.54	**0.64**	0.55	0.63	0.56	0.59	0.56	0.52
	MLP	**0.69**	**0.73**	**0.66**	0.61	**0.64**	0.61	**0.69**	**0.66**	**0.66**	**0.60**	**0.59**	**0.53**

On a broad analysis of the results obtained, a first point to highlight is that the proposed note tracking strategy achieves its best performance when considering ground-truth onset information (i.e., the one labelled as GT). While this may be seen as the expected behaviour, such results prove the validity of the note tracking method proposed: with the proper configuration (in this case, the most precise onset information that could be achieved for the data) this strategy is capable of retrieving performance values of $F_1 = 0.70$ in the frame-based analysis and $F_1 = 0.73$ in the note-based one. Note that such figures somehow constitute the maximum achievable performance of the proposed note tracking method given that actual onset estimators are not capable of retrieving such accurate onset description of a piece. Nevertheless, these values might be improved by considering the use of other descriptors different to the ones studied, obtained as either hand-crafted descriptors or with the use of *feature learning* approaches (e.g., Convolutional Neural Networks as in the work by Lee, Pham, Largman, and Ng (2009)) to automatically infer the most suitable features for the task.

When considering estimated onset events instead of ground-truth information there is a decrease in the performance of the note tracking system. In general, and as somehow expected, this drop is correlated with the goodness of the performance of the onset estimator. As a first example, SuF achieves the best results among all the onset estimators: its performance is, in general, quite similar to the case when ground-truth onset information is considered and only exhibits particular drops that, in the worst-case scenario, reach a value of 3 % and 10 % for the frame-based and note-based metrics, respectively, lower than the maximum achievable performance. The SD and CDD estimators exhibit a very similar performance between them, with the latter algorithm occasionally outperforming the former one; both estimators show a decrease between 3 % and 6 % for the frame-based metric and between 10 % and 20 % for the note-based figure of merit when compared to the ground-truth case. As of the SFB algorithm, while reported as the one achieving the lowest performance in terms of onset accuracy, it achieves very accurate note tracking figures that practically do not differ to the ones achieved by the SD and CDD algorithms. Finally, the use of random values for the onsets show the worst performance of all the onset descriptions. This behaviour is the expected one as the values used as onsets do not actually represent the real onsets in the audio signal.

Regarding the classification schemes, it may be noted that the eventual performance of the system is remarkably dependent on the classifier considered. Attending to figures obtained, the best results are obtained when considering an MLP as classifier, and occasionally an SVM scheme. For instance, in the ground-truth onset information case, MLP reports performance figures of $\mathrm{F}_1 = 0.70$ for the frame-based evaluation and $\mathrm{F}_1 = 0.73$ in the note-based one, thus outperforming all other classification strategies considered that also employ the same onset information. As accuracy in the onset information degrades, the absolute performance values suffer a drop (for instance, $\mathrm{F}_1 = 0.66$ in the note-based evaluation for the SuF estimator or $\mathrm{F}_1 = 0.60$ for the same metric and the CDD estimator), but MLP still obtains the best results. As commented the only strategy outperforming MLP is SVM for the particular cases when onset information is estimated with the SD and SFB methods and assessing with the onset-based metric. Nevertheless, experiments report that convergence in the training stage for the SVM classifier turns out to be much slower than for the MLP one.

On the other extreme, AB and DT generally report the lowest performance figures for the frame-based and note-based assessment strategies, respectively. For instance, in the ground-truth onset information case, AB reports a decrease in the frame-based metric close to 16 % with respect to the maximum reported by the MLP. Similarly, when compared to the maximum, DT reports a decrease close to a 20 % in the note-based assessment.

The NN classifier exhibits a particular behaviour to analyse. As it can be seen, this scheme retrieves fairly accurate results for both the frame-based and note-based metrics for the ground-truth onset information (on a broad picture, close to $\mathrm{F}_1 = 0.65$). Nevertheless, when another source of onset estimation is considered, the note-based metric degrades while the frame-based metric keeps relatively steady. As the NN rule does not perform any explicit generalisation over the training data, it may be possible that instances with similar feature values may be labelled with different classes and thus confuse the performance of the system.

The RaF ensemble-based scheme, while not reporting the best overall results, achieves scores that span up to values of $\mathrm{F}_1 = 0.68$ and $\mathrm{F}_1 = 0.70$ for the frame-based and note-based metric, respectively, with ground-truth onset information. While it might be argued that these figures may be improved by considering more complex classifiers, ensemble methods have been reported to achieve their best performance using simple decision schemes, such as the one-level decision trees used in this work. Given the simplicity of the classifiers, the convergence of the training model in RaF is remarkably fast, thus exhibiting an additional advantage to other classification schemes with slower training phases.

According to the obtained results, the use of additional features which consider the surrounding instances leads to different conclusions depending on the evaluation scheme considered. Except for the case when considering ground-truth onset information in which such information shows a general improvement in the performance of the system, no clear conclusions can be gathered when considering these additional features for the rest of the cases. For instance, consider the case of the SVM classifier with the SD estimator; in this case, note-based performance decreases from $F_1 = 0.61$ when no additional features are considered to $F_1 = 0.58$ when only the instances directly surrounding the one at issue are considered; however, when the information of two instances per side is included, the performance increases to $F_1 = 0.62$.

With respect to comparison with existing note tracking methods from the literature, Table 4 shows results in terms of F_1 comparing the following approaches: *Base*, which stands for the initial binarization of the posteriorgram, Poliner and Ellis (2007), using a two-stage HMM for note tracking, and Sigtia et al. (2016) which considers a music language model (MLM) based post-processing scheme. Finally *Classification* shows the best figures obtained with the proposed method for the different onset estimators. These methods are denoted by the same acronyms used previously in the analysis while ground-truth onset information is referred to as GT.

Table 4: Note tracking results on the MAPS dataset in terms of F_1, comparing the proposed method with benchmark approaches. *Base* stands for the initial binary frame-level transcription obtained; Poliner and Ellis (2007) refers to the HMM-based note tracking method proposed on that paper; Sigtia et al. (2016) represents the MLM-based post-processing technique; *Classification* stands for the proposed method with the different onset detection methods considered.

	Base	Poliner and Ellis (2007)	Sigtia et al. (2016)	Classification					
				GT	SD	SFB	SuF	CDD	Random
Frame	0.57	0.59	0.65	0.70	0.66	0.65	0.69	0.66	0.61
Note	0.62	0.65	0.66	0.73	0.62	0.64	0.68	0.62	0.61

As can be seen from Table 4, the proposed classification-based method stands as a competitive alternative to other considered techniques. For both frame-based and onset-based metrics, the proposed method is able to surpass the baseline approach by more than +10 % in terms of F_1 for both metrics considered.

When compared to the HMM-based method by Poliner and Ellis (2007), the proposed approach also demonstrates an improvement of +10 % when considering frame-based metrics and +3 % in terms of note-based metrics, when using the SuF onset detector. As expected, the improvement increases further when using ground truth onset information with the proposed method.

The method by Sigtia et al. (2016) achieves similar figures to the HMM-based approach with a particular improvement on the frame-based metric. In this sense, conclusions gathered from the comparison are quite similar: the proposed approach shows an improvement using frame-based metrics while for the note-based ones it is necessary to consider very precise onset information (e.g. the SuF method or the ground-truth onset annotations).

Finally, the existing gap between the figures obtained when considering ground-truth onset information and the SuF onset detector suggests that there is still room for improvement simply by focusing on improving the performance of onset detection methods.

5.2.1 Statistical significance analysis

Concerning statistical significance of the performance of the proposed note tracking approach compared to baseline (and random) approaches, the recognizer comparison technique of Guyon, Makhoul, Schwartz, and Vapnik (1998) is used. We compare pairs of note tracking methods, with the hypothesis that the difference between the two is statistically significant with 95 % confidence ($\alpha = 0.05$). The multi-pitch detection errors are assumed to be independent and identically distributed. Statistical significance experiments are not made on the level of each music piece (where each music piece can potentially contain hundreds of notes), but on the level of a musical note or a time frame, when using note-based and frame-based metrics, respectively. This is motivated by Benetos (2012) where statistical significance tests on multi-pitch detection on the level of a music piece were considered to be an oversimplification.

When considering the results from Table 3, the aforementioned tests indicate that for each window configuration and metric type, the best performing onset estimator significantly outperforms random onsets (in fact, even a 1 % difference in terms of F-measure can be considered significant given the dataset size). When considering the comparative note tracking results from Table 4, again it is observed that the proposed approach using the best performing onset detector (in this case, the SuF one) significantly outperforms both the baseline and comparative note tracking approaches.

6 Conclusions and future work

Note tracking constitutes a key process in Automatic Music Transcription systems. Such process aims at retrieving a high-level symbolic representation of the content of a music piece out of its frame-by-frame multipitch analysis. The vast majority of note tracking approaches consist of a collection of hand-crafted rules based on note-pruning and gap-filling policies particularly adapted to the data at issue.

In this paper we explored the use of a data-driven approach for note tracking by modelling the task as a supervised classification problem. The proposed method acts as a post-processing stage for an initial frame-level multi-pitch detection: each pitch band of the initial frame-level transcription is segmented into instances using onset events estimated from the piece and a set of features based on the multi-pitch analysis; each instance is classified as being an active or inactive element of the transcription (binary classification) by comparing to a set of labelled instances. Results obtained prove that the proposed approach is capable of outperforming other benchmark note tracking strategies as for instance a set of hand-crafted rules or the *de-facto* standard approach of a pitch-wise two-state Hidden Markov Model.

In sight of the results obtained, several lines of future work are further considered. As a first point to address we are interested in the study of the influence of the considered features in the overall performance of the system (e.g., considering feature selection methods) and in the further use of *feature learning* methods for the automatic estimation of new sets of descriptors for the systemas, for instance, with the use of Convolutional Neural Networks. Another point to address is the study of the performance of the proposed method in other timbres to assess its generalisation capabilities. Additional improvements may be observed if considering offset events, and hence we also consider it as a path to explore. Also, the further study of alternative descriptors to the ones proposed may give additional insights about the performance of the method. Finally, a last point that arises from this work is the possible exploration of improving the performance of HMM-based note tracking systems by including the temporal segmentation provided by the onset analysis of the piece.

References

Bay, M., Ehmann, A. F., & Downie, J. S. (2009, October). Evaluation of Multiple-F0 Estimation and Tracking Systems. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)* (pp. 315–320). Kobe, Japan.

Bello, J. P., Daudet, L., Abdallah, S. A., Duxbury, C., Davies, M. E., & Sandler, M. B. (2005). A Tutorial on Onset Detection in Music Signals. *IEEE Transactions on Speech and Audio Processing*, *13*(5), 1035–1047.

Bello, J. P., Daudet, L., & Sandler, M. B. (2006). Automatic piano transcription using frequency and time-domain information. *IEEE Transactions on Audio, Speech, and Language Processing*, *14*(6), 2242–2251.

Benetos, E. (2012). *Automatic transcription of polyphonic music exploiting temporal evolution* (Unpublished doctoral dissertation). Queen Mary University of London.

Benetos, E., & Dixon, S. (2011). Polyphonic music transcription using note onset and offset detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (pp. 37–40).

Benetos, E., & Dixon, S. (2013). Multiple-instrument polyphonic music transcription using a temporally constrained shift-invariant model. *The Journal of the Acoustical Society of America*, *133*(3), 1727–1741.

Benetos, E., Dixon, S., Giannoulis, D., Kirchhoff, H., & Klapuri, A. (2012, October). Automatic Music Transcription: Breaking the Glass Ceiling. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR).* Porto, Portugal.

Benetos, E., Dixon, S., Giannoulis, D., Kirchhoff, H., & Klapuri, A. (2013). Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, *41*(3), 407–434.

Benetos, E., & Weyde, T. (2015). An efficient temporally-constrained probabilistic model for multiple-instrument music transcription. In *Proceedings of the 16th International*

Society for Music Information Retrieval Conference (ISMIR) (pp. 701–707). Málaga, Spain.

Berg-Kirkpatrick, T., Andreas, J., & Klein, D. (2014). Unsupervised transcription of piano music. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 1538–1546).

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* New York, New York, USA: Springer-Verlag.

Böck, S., Krebs, F., & Schedl, M. (2012). Evaluating the Online Capabilities of Onset Detection Methods. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)* (pp. 49–54). Porto, Portugal.

Böck, S., & Widmer, G. (2013a, November). Local Group Delay based Vibrato and Tremolo Suppression for Onset Detection. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)* (pp. 589–594). Curitiba, Brazil.

Böck, S., & Widmer, G. (2013b, September). Maximum Filter Vibrato Suppression for Onset Detection. In *Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13)* (pp. 55–61). Maynooth, Ireland.

Cañadas-Quesada, F. J., Ruiz-Reyes, N., Vera-Candeas, P., Carabias-Orti, J. J., & Maldonado, S. (2010). A Multiple-F0 Estimation Approach Based on Gaussian Spectral Modelling for Polyphonic Music Transcription. *Journal of New Music Research*, *39*(1), 93-107.

Cheng, T., Dixon, S., & Mauch, M. (2015). Improving piano note tracking by HMM smoothing. In *Proceedings of the 23rd European Signal Processing Conference (EUSIPCO)* (pp. 2009–2013).

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B*, *39*(1), 1–38.

Dessein, A., Cont, A., & Lemaitre, G. (2010). Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)* (pp. 489–494).

Duan, Z., Pardo, B., & Zhang, C. (2010). Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions. *IEEE Transactions on Audio, Speech, and Language Processing*, *18*(8), 2121–2133.

Duan, Z., & Temperley, D. (2014, October). Note-level Music Transcription by Maximum Likelihood Sampling. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)* (pp. 181–186). Taipei, Taiwan.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification.* John Wiley & Sons.

Duxbury, C., Bello, J. P., Davies, M., & Sandler, M. (2003). Complex Domain Onset Detection for Musical Signals. In *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx)* (pp. 90–93). London, UK.

Emiya, V., Badeau, R., & David, B. (2008). Automatic transcription of piano music based on HMM tracking of jointly-estimated pitches. In *Proceedings of the 16th European Signal Processing Conference (EUSIPCO)* (pp. 1–5).

Emiya, V., Badeau, R., & David, B. (2010). Multipitch Estimation of Piano Sounds Using a New Probabilistic Spectral Smoothness Principle. *IEEE Trans. Audio Speech Lang. Process.*, *18*(6), 1643–1654.

Grindlay, G., & Ellis, D. (2011). Transcribing Multi-Instrument Polyphonic Music With Hierarchical Eigeninstruments. *Journal of Selected Topics in Signal Processing*, *5*(6), 1159–1169.

Grosche, P., Schuller, B., Müller, M., & Rigoll, G. (2012). Automatic transcription of recorded music. *Acta Acustica united with Acustica*, *98*(2), 199–215.

Guyon, I., Makhoul, J., Schwartz, R., & Vapnik, V. (1998, Jan). What size test set gives good error rate estimates? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(1), 52-64. doi: 10.1109/34.655649

Iñesta, J. M., & Pérez-Sancho, C. (2013). Interactive multimodal music transcription. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (pp. 211–215).

Klapuri, A., & Davy, M. (2007). *Signal processing methods for music transcription.* Springer Science & Business Media.

Kroher, N., Díaz-Báñez, J.-M., Mora, J., & Gómez, E. (2016). Corpus cofla: a research corpus for the computational study of flamenco music. *Journal on Computing and Cultural Heritage (JOCCH)*, *9*(2), 10.

Lee, H., Pham, P., Largman, Y., & Ng, A. Y. (2009, December). Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 1096–1104). Vancouver, Canada.

Lidy, T., Mayer, R., Rauber, A., Ponce de León, P. J., Pertusa, A., & Iñesta, J. M. (2010). A cartesian ensemble of feature subspace classifiers for music categorization. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)* (pp. 279–284). Utrecht, Netherlands.

Marolt, M., & Divjak, S. (2002). On detecting repeated notes in piano music. In *Proceedings of the 3rd International Society for Music Information Retrieval Conference (ISMIR)* (pp. 273–274).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pertusa, A., & Iñesta, J. M. (2012). Efficient methods for joint estimation of multiple fundamental frequencies in music signals. *EURASIP Journal on Advances in Signal Processing*, *2012*(1), 1–13.

Pertusa, A., Klapuri, A., & Iñesta, J. M. (2005, November). Recognition of Note Onsets in Digital Music Using Semitone Bands. In *Progress in pattern recognition, image analysis and applications: 10th iberoamerican congress on pattern recognition (ciarp)* (pp. 869–879).

Poliner, G., & Ellis, D. (2007). A discriminative model for polyphonic piano transcription. *EURASIP Journal on Applied Signal Processing*, *2007*(1), 154–154.

Raczyński, S. A., Ono, N., & Sagayama, S. (2009). Note detection with dynamic bayesian networks as a postanalysis step for NMF-based multiple pitch estimation techniques. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)* (pp. 49–52).

Ryynänen, M. P., & Klapuri, A. (2005). Polyphonic music transcription using note event modeling. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)* (pp. 319–322).

Sigtia, S., Benetos, E., & Dixon, S. (2016). An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*(5), 927-939.

Valero-Mas, J. J., Benetos, E., & Iñesta, J. M. (2016, September). Classification-based Note Tracking for Automatic Music Transcription. In *Proceedings of the 9th International Workshop on Machine Learning and Music (MML)* (pp. 61–65). Riva del Garda, Italy.

Valero-Mas, J. J., Benetos, E., & Iñesta, J. M. (2017). Assessing the relevance of onset information for note tracking in piano music transcription. In *Proceedings of the AES International Conference on Semantic Audio.*

Vincent, E., Bertin, N., & Badeau, R. (2010). Adaptive harmonic spectral decomposition for multiple pitch estimation. *IEEE Transactions on Audio, Speech, and Language Processing*, *18*(3), 528–537.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, *13*(2), 260–269.

Weninger, F., Kirst, C., Schuller, B., & Bungartz, H.-J. (2013). A discriminative approach to polyphonic piano note transcription using supervised non-negative matrix factorization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (pp. 6–10).