CONFIDENCE MEASURES AND THEIR APPLICATIONS IN MUSIC LABELLING SYSTEMS BASED ON HIDDEN MARKOV MODELS

Johan Pauwels **Ken O'Hanlon** **György Fazekas** **Mark B. Sandler**
Centre for Digital Music, Queen Mary University of London, UK
{j.pauwels, k.o.ohanlon, g.fazekas, mark.sandler}@qmul.ac.uk

ABSTRACT

Inspired by previous work on confidence measures for tempo estimation in loops, we explore ways to add confidence measures to other music labelling tasks. We start by reflecting on the reasons why the work on loops was successful and argue that it is an example of the ideal scenario in which it is possible to define a confidence measure independently of the estimation algorithm. This requires additional domain knowledge not used by the estimation algorithm, which is rarely available. Therefore we move our focus to defining confidence measures for hidden Markov models, a technique used in multiple music information retrieval systems and beyond. We propose two measures that are oblivious to the specific labelling task, trading off performance for computational requirements. They are experimentally validated by means of a chord estimation task. Finally, we have a look at alternative uses of confidence measures, besides those applications that require a high precision rather than a high recall, such as most query retrievals.

1. INTRODUCTION

Most of the efforts in music information retrieval research are directed towards improving the performance of various automatic labelling tasks. This consists of developing algorithms that are increasingly good at approximating some reference labels, often produced by human annotators, based on an input audio file. These labels represent different musical concepts, such as genre, tempo, instrumentation or musical key.

When such algorithms are deployed in real world scenarios, however, no explicit comparison is made between the generated labels and a reference. An example is the retrieval of audio based on musically meaningful search terms. The only relevant measure of performance here is the degree of satisfaction of the user with the returned audio files. The user will subconsciously verify if the returned audio corresponds somewhat to the query term, and be dissatisfied if it doesn't, but this implicit and informal evaluation is nowhere as rigorous as the numerical evaluation performed to demonstrate algorithmic improvements. This gap between algorithmic evaluation and user evaluation makes that increases in algorithmic performance do not necessary lead to increases in user satisfaction.

Crucially, in many retrieval tasks the precision is more important than the recall. The users only judge the quality of the returned audio, the amount of potentially useful audio files that are not returned to them are unknown and irrelevant (once the amount of returned files reaches a minimally acceptable number of course). A relatively easy way to improve the perceived quality of the returned audio (and thereby user satisfaction) would be to only return those files for which the generated labels are known to be of a high quality. This necessitates a reliable measure of confidence for the generated output labels, which must be calculated without relying on a known reference output.

Despite its obvious use-case, not much work has been performed on confidence measures for music labelling. For tempo estimation in music loops specifically, a thorough study has been recently performed by Font and Serra [5]. They propose a new confidence measure and compare it to earlier efforts of Zapata et al. [18]. In this paper, we devise new methods for confidence estimation that are not specific to a single task, but work with all systems based on hidden Markov models (HMMs). To this end, we start by analysing the reasons why the work on loops was successful and what we can and cannot reuse from it in Section 2. Then we look at the general framework of HMMs, propose some candidate confidence measures and evaluate them for chord estimation in Section 3. Next, a novel application for confidence measures is discussed in Section 4. We end by formulating some conclusions and directions for future work in Section 5.

2. DOMAIN-BASED VERSUS ALGORITHM-BASED CONFIDENCES MEASURES

To aid coming up with confidence measures for a wider range of tasks, it is useful to first reflect on the underlying conditions that made Font and Serra's work [5] successful. They managed to define a confidence measure that can be calculated from just the generated output. It is therefore oblivious to the algorithm that was used to calculate the output. The advantage is that knowledge of and

© Johan Pauwels, Ken O'Hanlon, György Fazekas, Mark B. Sandler. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Johan Pauwels, Ken O'Hanlon, György Fazekas, Mark B. Sandler. "Confidence measures and their applications in music labelling systems based on hidden Markov models", 18th International Society for Music Information Retrieval Conference, Suzhou, China, 2017.

access to the inner workings of the algorithm are not required in order to use the confidence measure. However, this type of confidence measure relies on extra prior knowledge about the application domain, which is used to verify the output against in the absence of internal states of the algorithm and (obviously) the target labels. Therefore we call these confidence measures *domain-specific*, as opposed to *algorithm-specific* measures. For example, the domain knowledge used for loops is that they are cut in such a way that each loop contains an exact number of beats.

It is of the utmost importance that this domain information hasn't been exploited yet by the estimation algorithm. All tested software in [5] fulfils this criterion, as they were developed for music pieces in general, not just loops. If algorithms would already rely on this prior knowledge, the output would be internally adjusted by keeping only those tempo candidates that lead to an exact multiple of beats for the duration of the loop. The confidence measure would then always be maximal and therefore useless.

Finding such unexploited knowledge for a specific application is not always possible, and if there is one, it also needs to be distinctive enough. Take for instance the case of key estimation in loops. A reasonable prior would be to assume that there are no key changes for the duration of a loop. Even if a key estimation algorithm is capable of producing key changes, it is unlikely that multiple keys will be generated over the short duration of a typical loop. No reliable confidence measure can then be derived from this additional information.

We argue that having sufficient unexploited and distinctive domain knowledge for a particular task is a rare event. In practice, it is therefore more likely that we need recourse to algorithm-specific confidence measures. These are defined using the intermediate states of the algorithm, which unfortunately means that separate measures need to be formulated for each algorithm and that the resulting confidence cannot easily be compared between algorithms. The upside is that they are not tied to a particular domain.

In the remainder of this paper, we propose some candidate algorithm-based confidence measures. To mitigate their algorithm-specificity, we will look at the framework of hidden Markov Models [15], which is commonly used in a variety of estimation tasks. Our hope is that the proposed solutions will therefore be task-independent and widely applicable.

3. HIDDEN MARKOV MODEL-BASED CONFIDENCE MEASURES

3.1 Hidden Markov Model Basics

According to Ghahramani [7], "a hidden Markov model (HMM) is a tool for representing probability distributions over sequences of observations." It is widely used to model sequences in applications as diverse as speech recognition [15] and bioinformatics [4]. In music information retrieval, it is commonly used to take temporal dependencies into account when observations are localised.

Formally, an HMM is a doubly stochastic process that consists of a first-order Markov chain of hidden states that can only be observed through another, visible stochastic process. Both processes are sampled at discrete intervals, so they can be represented by an index variable t. This sequence index often represents time. The observed variables can be discrete or continuous, finite or infinite, univariate or multivariate, or any combination thereof, as long as they have a probability distribution associated with them. The state variables, on the other hand, are always discrete and there's a finite number N of them. The values the state variable can possibly take are therefore enumerated as $S_n, \forall 1 \leq n \leq N$. The value of the specific hidden state at index t is represented as Y_t, so $Y_t \in \mathbf{S} = \{S_1, \ldots, S_N\}$. The observed variable at index t is represented as X_t. It can potentially take an uncountable number of values, so we can't enumerate them, only represent their space by $\mathbf{O}$. Furthermore, an observed variable X_t is assumed to depend only on the hidden state Y_t at the corresponding position t in the sequence, not on the hidden or observed variables at any other positions.

Hidden Markov models are entirely described by specifying three probability distributions: (1) the initial state distribution; (2) the state transition distribution, which is time-invariant in a standard HMM; (3) the observation distribution, which is also time-invariant in a standard HMM

$$P\left(Y_1 = S_i\right) \equiv \pi_i, \forall 1 \leq i \leq N \tag{1}$$

$$P\left(Y_{t+1} = S_j | Y_t = S_i\right) \equiv a_{ij}, \forall 1 \leq i, j \leq N \tag{2}$$

$$p\left(X_t = O | Y_t = S_j\right) \equiv b_j\left(O\right), \forall 1 \leq j \leq N, \forall O \in \mathbf{O} \tag{3}$$

The set of parameters of an HMM can therefore be summarised as $\lambda = \{\Pi, A, B\}$, where $\Pi = \{\pi_i\}_i$, $A = \{a_{ij}\}_{i,j}$, $B = \{b_j\left(O\right)\}_{j,\mathbf{O}}$

The context in which confidence measures are useful assumes that the HMM parameters λ are already determined. Given a particular sequence of observations $x^{1:T} = x_1, \ldots, x_T$, we want to determine the underlying state sequence (called path) $y^{1:T} = y_1, \ldots, y_T$ that produced these observations and a value c that indicates how much confidence we have in the generated hidden state sequence. The process that determines the optimal hidden state sequence is called "decoding" the HMM and is well established in the literature [15]. Our goal is to find out which of the internal states of the decoding process could be repurposed as a confidence measure.

The most common way to decode an HMM is to find the single path $\hat{y}^{1:T}$ that is the *maximum a posteriori* (MAP) estimate:

$$\hat{y}^{1:T} = \underset{y^{1:T} \in \mathbf{S}^T}{\operatorname{argmax}}\, p\left(y^{1:T} | x^{1:T}, \lambda\right) \tag{4}$$

$$= \underset{y^{1:T} \in \mathbf{S}^T}{\operatorname{argmax}}\, p\left(y^{1:T}, x^{1:T} | \lambda\right) \tag{5}$$

This path can be found by following the Viterbi-algorithm [17]. For more details about its implementation, we refer to Rabiner's well-known tutorial [15].

3.2 Experimental Setup

In order to experimentally validate the theoretical confidence measures we're about to propose, we need a concrete labelling system based on an HMM. In this section, we describe the chord estimation system that will be used to this end.

Our system first converts an audio file into mono and extracts a time-chroma [6] representation from it that will be used as observations in the HMM. We use two different chroma extractors, such that we can verify that the confidence measure works regardless of features. The first variant we use is recently developed by Korzeniowski et al. [9]. They trained a three layer dense neural network to map quarter-tone log-frequency magnitude spectra to chromas. The second variant is the Compressed Log Pitch (CLP) chroma [12], which applies logarithmic compression before summing the semi-tone log-frequency power spectra into one octave. Both implementations are taken from the *madmom* library [2], version *0.15.1*. The parameters were set to the values proposed in their original papers.

Because the features that are fed into the HMM are chromas, this means the observation space consists of twelve-dimensional positive real numbers $\mathbf{O} = \mathbb{R}_{\geq 0}^{12}$. The observation probability distribution $b_j(O)$ over $\mathbf{O}$ is calculated using template matching. This requires that each chord state S_j has a template M_j associated with it and a similarity measure that maps observation-template pairs (O, M_j) to probabilities. We use the normalised cosine similarity, defined as

$$b_j(O) = \frac{\langle O, M_j \rangle}{\|O\|_{L2} \, \|M_j\|_{L2}} \tag{6}$$

where $\langle O, M_j \rangle$ represents the inner product between O and M_j.

In our experiments, we set the number of chord states to 48. We discern four chord types (maj-min-dim-aug triads) for each of the twelve possible roots. The associated chord templates are binary, with the chromas that are theoretically present in the chord set to one and the other chromas set to zero.

The last parts of the HMM that need to be configured are the initialisation and the transition probabilities. Both systems are initialised uniformly, i.e. the probability to start in a specific state is $1/48$ for all chords. The transition probabilities are kept deliberately simple. The state self-transition probabilities are all assigned the same value

$$a_{ii} = \tau, \forall 1 \leq i \leq N \tag{7}$$

whereas the state-changing probabilities are distributed uniformly

$$a_{ij} = \frac{1-\tau}{N-1}, \forall 1 \leq i, j \leq N, i \neq j \tag{8}$$

This reduces the number of parameters considerably, thereby reducing the potential that our experiments don't translate to other datasets, but it has also been demonstrated that such a simple transition matrix is enough to get most of the benefits of applying an HMM to the observations. In [13], it is shown that the state-changing probabilities in an HMM where states represent relative chords in a key can at most improve the estimation performance by 2–3 %-points, whereas [3] show that this is even less when states represent absolute chords, without the context of a key (as is the case here too). The HMM then effectively acts as a probabilistic temporal smoother, and does not take into account the specific values of surrounding states, only their duration. The only remaining parameter τ is determined through an exhaustive search on the test data.

The score that would ideally be predicted by the confidence measure is calculated by the open-source MusOOEvaluator [1] tool [14]. This is the same software that is used for MIREX. We use the "MirexMajMin" preset for chord evaluation.

Finally, we use two datasets for testing the chord estimation systems and their confidence measures, in order to investigate data-specific behaviour. The first is the "Isophonics" [2] dataset [11]. Specifically, we use the subset that is used for the MIREX chord estimation task. It contains 217 songs and is comprised of 12 Beatles albums (180 songs), a Queen compilation (19 songs) and one Zweieck album (18 songs). The second is the "RWC Popular" [3] dataset [8]. The latter contains 100 Japanese pop songs purposefully recorded for music information retrieval research.

3.3 Sequence Probability as Confidence Measure

As part of the MAP decoding, the probability of the optimal label sequence gets returned:

$$p\left(\hat{y}^{1:T}\right) = \max_{y^{1:T} \in \mathbf{S}^T} p\left(y^{1:T} | x^{1:T}, \lambda\right) \tag{9}$$

$$= \max_{y^{1:T} \in \mathbf{S}^T} p\left(y^{1:T}, x^{1:T} | \lambda\right) \tag{10}$$

Although this seems like an obvious candidate for a confidence measure, as far as we know, nobody has ever examined the correlation between optimal path probability and labelling score. We know from the definition that the optimal path probability has the highest value relative to the probabilities of any other paths, but in order for it to be useful as a confidence measure, its absolute value matters.

Since the joint probability $p\left(y^{1:T}, x^{1:T}\right)$ can be decomposed as

$$p\left(y^{1:T}, x^{1:T}\right) = P(y_1)\, p(x_1|y_1) \prod_{t=2}^{T} P(y_t|y_{t-1})\, p(x_t|y_t) \tag{11}$$

a first step that needs to be taken is to normalise the path probability with respect to the song duration T, as $p\left(y^{1:T}, x^{1:T}\right)$ gets progressively smaller with T [4]. This

[1] https://github.com/jpauwels/MusOOEvaluator

[2] annotations available at http://isophonics.net/content/reference-annotations

[3] annotations available at https://github.com/tmc323/Chord-Annotations

[4] In practice, we're working in the logarithmic domain precisely to mitigate this vanishing probability problem

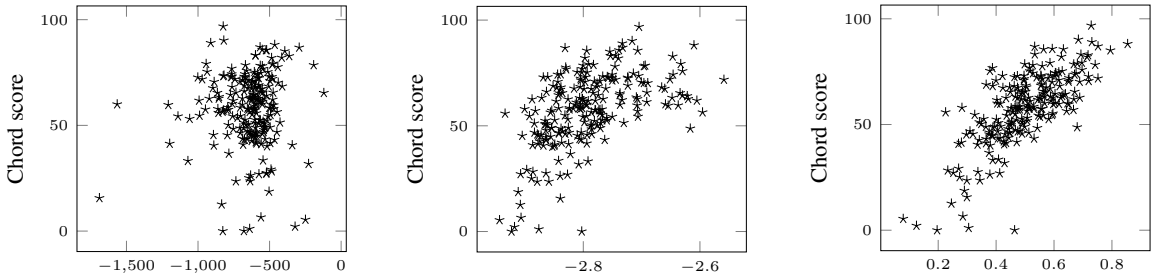

Figure 1: Chord estimation scores for CLP features on the Isophonics dataset: as a function of (a) $\log\left(\sqrt[T]{p\left(\hat{y}^{1:T}\right)}\right)$, (b) $\mathrm{median}_t\left(\log\left[P\left(y_t|y_{t-1}\right)p\left(x_t|y_t\right)\right]\right)$, (c) the pointwise path difference

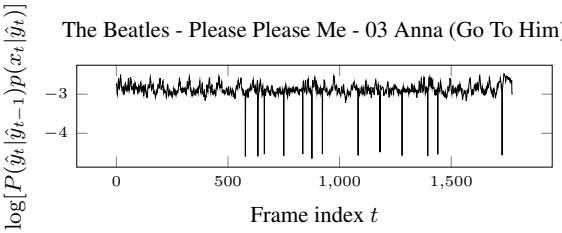

Figure 2: Separate contributions to the optimal path probability $\log\left[P\left(y_t|y_{t-1}\right)p\left(x_t|y_t\right)\right]$ for every frame t

way, we can compare songs of different length. Because the probabilities per frame get accumulated through a product, it makes sense to take the T-th root of the path probability. Then we are effectively calculating the geometric mean of the contributions per frame to the overall path probability.

We've plotted in Figure 1a an example of the scores as a function of the resulting geometric mean for the system with CLP features run on the Isophonics dataset. It is immediately clear from the figure that there is little correlation between the two axes and therefore the optimal path probability is not suitable as a confidence measure.

To find out why this is the case, we take a look at the individual contributions per frame $P\left(\hat{y}_t|\hat{y}_{t-1}\right)p\left(x_t|\hat{y}_t\right), \forall t$ separately. In Figure 2, we show these probabilities for an example song. We can see that there are some strong outliers in these probabilities, and this is the case for every song, not just this example. Because the contributions per frame are multiplied [5] to form the overall probability, we postulate that the limited number of outliers dominate the overall probability regardless of the performance on other frames, such that there is no longer a relation between the overall probability and how well $\hat{y}^{1:T}$ explains $x^{1:T}$. Note that the presence of such an outlier at frame t does not necessarily mean that the global optimal path strays from the locally optimal path at that frame. It could also mean that none of the states can explain that particular observation well $b_j\left(x_t\right) \approx 0, \forall j$. Cases like this are not problematic for the determination of the globally optimal path, since only the difference between observation distributions per state is relevant for the Viterbi algorithm, but the overall probability will be affected.

Since the outliers of $\left(P\left(y_t|y_{t-1}\right)p\left(x_t|y_t\right)\right)$ prove to be so problematic, it makes sense to aggregate the frames differently than through a geometric mean. Instead, we take the median, such that the exact magnitude of the lowest probabilities doesn't matter. We plotted the results in Figure 1b for the same chord estimator configuration as Figure 1a. The figure shows a marked improvement with respect to the geometric mean based confidence measure.

We do believe that it should be possible to achieve a clearer linear relationship between score and confidence measure though. However, as far as repurposing the internal variables of the standard Viterbi algorithm go, we feel we have reached a limit. The advantage of the median-based confidence measure is that it requires practically no more computation time than standard MAP decoding. It only requires the so-called lattice of intermediate path probabilities $p\left(x^{1:t}, y^{1:t}, y_t = S_i|\lambda\right), \forall i, t$ to be kept in its entirety, as opposed to only needing to keep the previous and current frame (t and $t-1$), which obviously leads to an increase in memory. In the next sections, we will compare this measure with a more computationally expensive one and report on more thorough experiments.

3.4 Combining Decoders as Confidence Measure

While the Viterbi algorithm returns the globally optimal label sequence in the MAP sense of the word, the definition of "optimal" is inherently ambiguous. Another criterion of optimality leads to another decoding method. A common alternative to MAP decoding is *pointwise maximum a posteriori (PMAP)* decoding [6]. If we represent the path estimate returned through PMAP decoding from now on as $\tilde{y}^{1:T}$, then we find

$$\tilde{y}_t = \underset{y_t \in \mathbf{S}}{\operatorname{argmax}}\, p\left(y_t|x^{1:T}, \lambda\right) \tag{12}$$

As the name implies, the optimal path is determined point-by-point in such a way that the expected number

[5] The values depicted in Figure 2 are actually summed, because we work in the logarithmic domain.

[6] Confusingly, this decoding algorithm is known under many names, posterior decoding or max-gamma decoding just a few of them. More alternative names can be found in [10, p. 4].

of correct individual states is maximised. The probability to be in a given state s at frame t is found by means of helper forward $\alpha_t(s)$ and backward $\beta_t(s)$ variables obtained through a process known as the forward-backward algorithm [1]:

$$p\left(s|x^{1:T}\right) = \frac{\alpha_t(s)\,\beta_t(s)}{\sum_{r\in\mathbf{S}}\alpha_t(r)\,\beta_t(r)} \tag{13}$$

For further implementation details, we again refer to [15].

It depends strongly on the application whether MAP or PMAP decoding will lead to the best results [7] and to which extent they produce different paths. However, it is known that MAP decoding is most effective when a single path through the HMM strongly dominates all other ones, whereas PMAP decoding gives better results when multiple competing paths have similar overall probabilities [4].

We postulate that when the two methods of decoding yield the same path, this gives a good indication that the path will have a good score. Thus we derive a new confidence measure $\mathrm{PPD}\left(y^{1:T}\right)$ based on the pointwise path differences (PPD) as

$$\mathrm{PPD}\left(y^{1:T}\right) = \frac{1}{T}\sum_{t=1}^{T}\delta\left(\hat{y}_t, \tilde{y}_t\right) \tag{14}$$

where δ is the Kronecker-delta, and $\hat{y}_t$/$\tilde{y}_t$ the MAP/PMAP path estimates at time index t.

To illustrate the PPD, we take once more the same HMM configuration as in Figure 1a and plot the score as a function of the PPD in Figure 1c. Here the relation between the two is more clearly linear. The drawback of this confidence measure is obviously that it requires an additional PMAP decoding step, which requires extra computational power. Some steps, such as the calculation of the observation probabilities, are the same for both decoding algorithms though, so there's potential for reuse. Note that we still return the same MAP path as before, because it gives better results for our HMM configurations than the PMAP path, but the latter is available too.

3.5 Evaluation

As we now have two candidate confidence measures, we can systematically test them on the four combinations of features and datasets. Therefore we perform a similar experiment as in [5]. We start by taking the duration-weighted average score over the complete dataset and then progressively filter the files by first excluding those for which the confidence measure is the lowest. A good confidence measure will then lead to a monotonic increase in score as the filtering threshold increases. The results for the Isophonics dataset and the RWC Popular dataset can be found in Figure 3a, respectively Figure 3b.

In all cases, we observe that the PPD is working well as a confidence measure. The score of the filtered dataset increases monotonically, save for a few exceptions when the number of remaining songs in the dataset becomes so low that the average scores become noisy. The curves of the filtered dataset size as a function of confidence cutoff are also close to straight, which means the PPD is nearly linearly distributed between its extrema. As expected, the median of the per frame contributions to the optimal path is less suitable as a confidence measure. The filtered score initially increases in all situations though, so it can still be used to remove the files with the lowest confidence from the dataset. Doing so will increase the precision when looking for a particular chord sequence in a dataset, for example, at the expense of decreasing the recall. Particularly for the CLP features, the median-based confidence measure seems to work less well. A possible explanation is that the neural network based chromas take context into account. The observations derived from the CLP features are therefore noisier, which will affect the median more.

4. OPTIMAL CHANNEL SELECTION BASED ON CONFIDENCE MEASURE

In this section, we explore an alternative usage for confidence measures. Traditionally, labels in music information retrieval are estimated from mixed down mono audio. Using the mono mix ensures that all information present in the audio is used for the label estimation. For certain types of labels however, it might be beneficial to selectively ignore some of the information. For example, ignoring percussion while estimating chords can be helpful, which has led to percussion separation techniques [16].

If we have multi-channel audio at our disposal, it is therefore possible that analysing a specific channel or combination of channels leads to higher quality labels than when the mono mixdown is used. The problem is then how to determine this (combination of) channel(s). Next, we'll verify if a confidence measure can be used for this.

Ideally, we'd use multi-channel or multi-track audio for this experiment, but since there is no such dataset with annotated chords, we propose an alternative. Starting from stereo Isophonics audio files, we demix the left (L) and right (R) channel according to their panning position into centre (C), hard left (HL) and hard right (HR). We employ the technique used by the "center cut" [8] audio filter of the open-source video editor VirtualDub. It operates in the complex spectral domain and relies on the fact that HL and HR are perpendicular to each other, such that L = C + HL and R = C + HR. In addition to these channels, we calculate the mono (L + R) and sides (HL + HR), such that we end up with seven virtual channels per song.

For each channel, we estimate the chord sequences from CLP features and their confidences. We first aim to determine the theoretical limits of optimal channel selection by performing an oracle-style experiment where we select the channels that lead to the biggest increase and biggest decrease in chord score when compared to the reference mono channel. Then we check how well we can retrieve the optimal channel by selecting the one that returns the chord sequence with the highest confidence.

[7] We tried both and verified that MAP decoding generally gives the best result for our proposed chord estimation system

[8] http://www.virtualdub.org/blog/pivot/entry.php?id=102

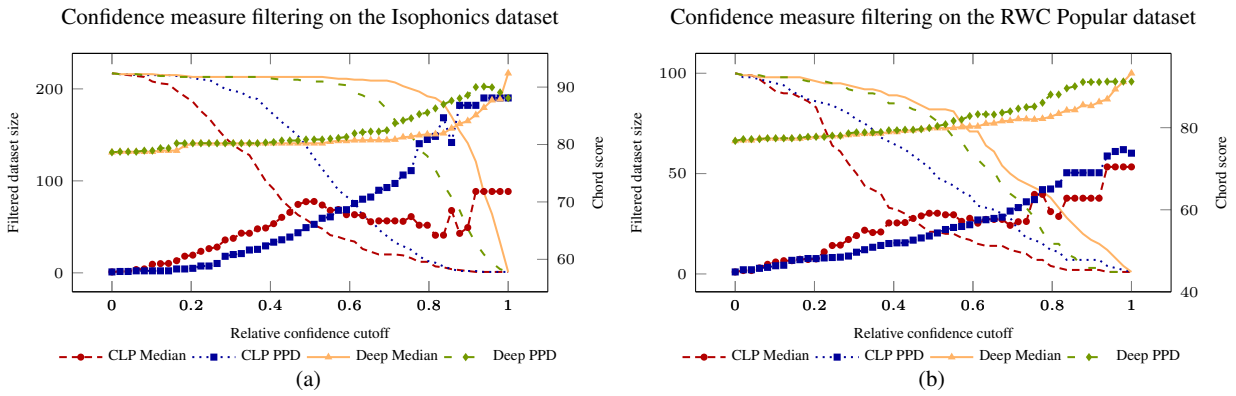

Figure 3: Confidence filtered chord scores for two datasets. The marks indicate the average score over the filtered dataset. The lines represent the number of remaining files in the database as a function of the confidence cutoff.

Album title	Mono score	Oracle best Δ	Oracle worst Δ	Median conf. Δ	PPD conf. Δ
The Beatles - Please Please Me	52.86	6.01	-13.93	-7.23	1.17
The Beatles - With the Beatles	56.22	0.60	-26.81	-18.72	-2.11
The Beatles - A Hard Day's Night	56.21	3.05	-31.91	-21.88	2.24
The Beatles - Beatles for Sale	63.98	9.33	-3.26	5.91	7.33
The Beatles - Help!	52.54	11.82	-13.23	7.90	10.03
The Beatles - Rubber Soul	59.25	5.44	-18.89	-3.00	2.77
The Beatles - Revolver	66.06	5.64	-17.63	-0.70	3.71
The Beatles - Sgt. Pepper's Lonely Hearts Club Band	51.33	4.93	-19.81	-4.83	-1.74
The Beatles - Magical Mystery Tour	66.77	3.52	-18.33	-3.10	0.54
The Beatles - The Beatles (CD1)	62.45	6.32	-17.82	1.76	1.68
The Beatles - The Beatles (CD2)	52.05	6.16	-18.91	1.38	1.67
The Beatles - Abbey Road	63.86	8.39	-17.47	4.33	4.11
The Beatles - Let It Be	61.16	11.96	-8.09	9.24	8.21
Queen - Greatest Hits I	47.50	7.86	-3.54	6.53	6.66
Queen - Greatest Hits II	66.16	4.02	-5.49	-1.45	-1.50
Zweieck - Zwielicht	54.73	7.73	-8.10	5.35	6.26
Overall	57.81	6.60	-14.97	-0.39	3.42

Table 1: Channel selection results grouped per album, using CLP features. The reference mono channel score is reported along with the absolute score differences for the best and worst oracle-style and the confidence-based channel selection.

The channel selection results overall and per album are reported in Table 1. From the oracle experiments we learn that a sizeable improvement in chord score can potentially be achieved by selecting the optimal channel, but also that the consequences of choosing the wrong channel can be severe. The PPD measure can be used successfully to determine a better channel than the mono reference, and manages to get a bit more than half the theoretically maximal improvement. The median-based confidence measure, on the other hand, is not suitable to select the optimal channel.

Based on the individual results per album, no relation with mixing style can be established. The mixing practices range from the mono-like early Beatles albums to the hard-panned late Beatles albums, with more modern Queen and Zweieck in between, but no trends in the (potential) score increase can be identified. Note that when we repeated the experiments with the DeepChroma chord estimation system, the oracle-based maximal increase was barely over 2%-points, and the PPD increase proportionate. A reason might be that the neural network is trained on mono mixes.

5. CONCLUSION AND FUTURE WORK

In this paper, we investigated confidence measures for HMM-based music labelling systems. We formulated two measures, a simple one that doesn't require extra computational power and a better one that is more demanding. They were tested for their ability to filter low-quality output of a chord estimation system. Finally, the capacity of the confidence measures to select the most optimal channel to use for chord estimation has been evaluated.

We hope that the applicability of the proposed confidence measures to other labelling tasks will be verified in the future, by ourselves or by others. To improve the chance of the latter, we've created a new general and modular HMM library [9], usable with C++ and Python, that includes the proposed measures. The code specific to the chord estimation experiments and the presented figures can also be found on-line [10].

Further work will include investigating whether a confidence measure can be used to select the optimal HMM parametrisation. For instance, the self-transition probability τ is currently set to a dataset-wide optimal value, even though it is clearly related to harmonic rhythm and therefore song-dependent. It might be worth investigating if the best value out of a number of candidates can be selected based on confidence.

[9] https://github.com/jpauwels/Hiddini

[10] https://github.com/jpauwels/chord-estimation-confidence

6. ACKNOWLEDGEMENTS

This work has been partly funded by the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/L019981/1 and by the European Union's Horizon 2020 research and innovation programme under grant agreement N° 688382.

7. REFERENCES

[1] Leonard E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73(3):360–363, May 1967.

[2] Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer. madmom: a new python audio and music signal processing library. In *Proceedings of the 24th ACM International Conference on Multimedia*, pages 1174–1178. ACM, 2016.

[3] Taemin Cho and Juan P. Bello. On the relative importance of individual components of chord recognition systems. *IEEE Transactions on Audio, Speech and Language Processing*, 22(2):477–492, February 2014.

[4] Piero Fariselli, Pier Luigi Martelli, and Rita Casadio. A new decoding algorithm for hidden markov models improves the prediction of the topology of all-beta membrane proteins. *BMC bioinformatics*, 6(Suppl 4):S12, 2005.

[5] Frederic Font and Xavier Serra. Tempo estimation for music loops and a simple confidence measure. In *Proceedings of the 17th Conference of the International Society for Music Information Retrieval (ISMIR)*, pages 269–275, 2016.

[6] Takuya Fujishima. Realtime chord recognition of musical sound: a system using Common Lisp Music. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 464–467, 1999.

[7] Zoubin Ghahramani. An introduction to hidden Markov models and Bayesian networks. *International journal of pattern recognition and artificial intelligence*, 15(01):9–42, 2001.

[8] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. RWC music database: Popular, classical and jazz music databases. In *Proceedings of the 3rd International Symposium on Music Information Retrieval (ISMIR)*, volume 2, pages 287–288, 2002.

[9] Filip Korzeniowski and Gerhard Widmer. Feature learning for chord recognition: the deep chroma extractor. In *Proceedings of the 17th Conference of the International Society for Music Information Retrieval (ISMIR)*, 2016.

[10] Jüri Lember and Alexey A. Koloydenko. Bridging Viterbi and posterior decoding: a generalized risk approach to hidden path inference based on hidden Markov models. *Journal of Machine Learning Research*, 15(1):1–58, 2014.

[11] Matthias Mauch, Chris Cannam, Matthew Davies, Simon Dixon, Chris Harte, Sefki Kolozali, Dan Tidhar, and Mark Sandler. OMRAS2 metadata project 2009. In *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR)*, 2009.

[12] Meinard Müller and Sebastian Ewert. Chroma toolbox: Matlab implementations for extracting variants of chroma-based audio features. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, pages 215–220, 2011.

[13] Johan Pauwels and Jean-Pierre Martens. Combining musicological knowledge about chords and keys in a simultaneous chord and local key estimation system. *Journal of New Music Research*, 43(3):318–330, 2014.

[14] Johan Pauwels and Geoffroy Peeters. Evaluating automatically estimated chord sequences. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[15] Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[16] Yushi Ueda, Yuki Uchiyama, Takuya Nishimoto, Nobutaka Ono, and Shigeki Sagayama. HMM-based approach for automatic chord detection using refined acoustic features. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5518–5521, March 2010.

[17] Andrew J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, April 1967.

[18] Jose Ricardo Zapata, Matthew E. Davies, Andre Holzapfel, Joao L. Oliveira, and Fabien Gouyon. Assigning a confidence threshold on automatic beat annotation in large datasets. In *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR)*, pages 157–162, 2012.