“Copy and Scale” Method for Doing Time-Localized M.I.R. Estimation: Application to Beat-tracking

Geoffroy Peeters
IRCAM - CNRS
1, pl. Igor Stravinsky - 75004 Paris - France
peeters@ircam.fr

ABSTRACT

In this work we propose a ”copy and scale” method based on the 1-NN paradigm to estimate time-localized parameters and apply it to the problem of beat-tracking. The 1-NN algorithm consists in assigning the information of the closest item of a pre-annotated database to an unknown target. It can be viewed as a ”copy and paste” method. The “copy and scale” method we propose consists in ”scaling” this information to adapt it to the properties of the unknown target. For this, we first represent the content of an audio signal using a sampled and tempo-normalized complex DFT. This representation is used as the vectors over which the 1-NN search is performed. Along each vector of the 1-NN space, we store the corresponding annotated beat-marker positions in a normalized form. Once the closest vector is found, its tempo is assigned to the unknown item and the normalized beat-markers are scaled to this tempo in order to provide the estimation of the unknown item beat-markers. We perform a preliminary evaluation of this method and show that, with such a simple method, we can achieve results comparable to the ones obtained with sophisticated approaches.

Categories and Subject Descriptors

H.5 [**Information Interfaces and Presentation**]: Sound and Music Computing

General Terms

Algorithms

1. INTRODUCTION

Music Information Retrieval from audio signal can be roughly divided into two categories: - estimation of global parameters (global meaning that the parameters is applicable to the whole file duration, an example of this is the music genre) - and estimation of local parameters (local meaning that the parameters is time-localized, examples of this are beat/downbeat, chord or pitch). Problems of the first category are usually solved using machine-learning approaches including the K-NN method. For K equal 1 (1-NN method) the method can be viewed as a “copy and paste” method, where the parameters (the music genre or music mood class) of an unknown item are estimated by “copying and pasting” the parameters of the closest item of a pre-annotated database. Problems of the second category are usually solved using signal processing algorithm without machine-learning techniques. In this work we propose a ”copy and scale” method based on the 1-NN paradigm to estimate local (time-localized) parameters and apply it to the problem of beat-tracking.

Suppose we have a very large database of audio-items, each one of them has been annotated with beat positions. Suppose we want to annotate an unknown audio extract with beat positions. The usual process is to run a beat-tracking algorithm on it. However, one may think of using an audio fingerprint technique to look if this extract is present in the database, get the precise time position of it and then simply “copy and paste” the annotated beat-markers of the database to the unknown item. However, this would require a very large data-set and require that the unknown audio extract is part of the database items. However, instead of using an audio fingerprint techniques (which implies an exact match of timbre, harmony, instrument and production), we relax the code to only highlight one of the specific aspect of the content. For example, we define a code such that the distance between two audio items is small when they have the same rhythmic pattern, tempo and are time-aligned. Then any item in the database with a very small distance to the unknown item can be used to provide the beat-markers of it; even if it is different from the unknown item. The required database may still be very large. We further relax the constraint on the code to provide small distances when the “rhythmic pattern” are close; this independently of the tempi and time-alignments. Then the required database can be much smaller since we only require it to represent the diversity of rhythm patterns. Of course, because the matching is not anymore complete (maybe the closest item has a different tempo and/or alignment) it may be necessary to re-aligned and re-scale the beat-markers before copying them. The code must provide the necessary information for this.

Paper content and organization: Starting from this, we propose in this work, a method which allows to apply a K-NN approach (with $K = 1$) for the estimation of time-localized information and apply it to the case of beat-marker estimation. Instead of the usual “copy and paste” approach underlying the 1-NN approach, we propose a “copy and scale” approach.

MML’10, October 25, 2010, Firenze, Italy.

The above mentioned distance between two items is obtained by coding the items using a sampled and tempo-normalized complex DFT. This code is said to be tempo-independent, since any two audio items with similar rhythm pattern but different tempo will have the same code (see [11] for more details). This is obtained by normalizing the frequencies by the tempo frequency. The inclusion of the DFT imaginary part provides the information for the time-alignment between any two sequences (through phase relationships). We describe this representation in part 2.1 and 2.2. A K-NN database of annotated audio item is created. For each annotated item, we store its coded representation which will be used to perform the search. Along each code we store the time-localized M.I.R. information which will be used to "copy and scale" the estimation to the unknown item. We store the item's tempo, rhythm class and time-normalized markers. We explain this in part 2.3. For an unknown item, we then extract a set of codes corresponding to possible tempo assumptions. For each code, we perform a search in the K-NN database using a complex distance. We describe this in part 2.4. The tempo assumption and the database-item providing the smallest distance are chosen to provide the estimation of the unknown item parameters: tempo, rhythm class and beat-markers. We describe this in part 2.5. We perform an evaluation on the "ballroom-dancer" test-set using a Leave-One-Out approach. We describe this in part 3. We finally conclude in part 4.

Motivation for the present work: There exist numerous beat-tracking and tempo-estimation methods, all based on complex signal processing algorithms. Failures to detect correctly the tempo and beat markers are mainly caused by 1) weak onsets, 2) time-variable tempo and 3) complex rhythm patterns (such as Latin music, African music). The third cause is due to the lack of knowledge of these patterns. It is of course possible to introduce on a case by case this knowledge. The proposed approach may be though as an easy way to introduce this knowledge on a large scale without explicit models (K-NN does not use any models).

Related works: A large number of works related to tempo estimation, beat-tracking or rhythm classification have been published. We refer the reader to [5] or [9] for an overview of recent approaches and/or results. Works which are the most related to our work are the followings. In [4], Eronen proposes to use a database of templates and a K-NN-regression to find the best tempo of an unknown signal. However, he does not deal with beat-tracking in this paper. In [11], we show that rhythm classification can be achieved with a high accuracy using solely the observation of the normalized amplitude DFT. In [8] Grosche proposes to use the phase of DFT spectrum to derive the beat-positions. Although not directly related to our approach, we also mention the work of Gouyon [6] who use machine-learning to classify signal-frames into beat and non-beat classes. We didn't find any previous work concerning the use of K-NN for beat-tracking, or concerning K-NN in the complex domain. We therefore think that our proposed approach is novel.

2. PROPOSED APPROACH

2.1 Item representation

For a given audio item, we first extract an onset-energy-function $o(n)$ representing at each time the likelihood of an onset. The method used for this is explained in [10]. The function has a sampling rate of 200.45Hz. We perform a Short Time Fourier Transform analysis of $o(n)$ using a window length of 8s and hop size 1s. We denote it by $X_k(o, t_i)$, where i is the frame index and k is the index of the Fourier frequencies f_k. Considering a tempo frequency $b(t_i)$ (expressed in Hz) over time t_i, we sample the complex spectrum $X_k(o, t_i)$ at the frequencies $f_k = b(t_i) \cdot f_l$ with $f_l = \{l/4 : 1 \leq l \leq 32\} \cup \{l/3 : 1 \leq l \leq 24\}$. These frequencies represent the harmonic series corresponding to a 4/4 meter ($b(t_i)/4$) and a 3/4 meter ($b(t_i)/3$) up to $8b(t_i)$. We note $X_l(o, b(t_i), t_i)$ the 48-dimensional complex vector representing time t_i for a tempo $b(t_i)$. $X_l(o, b(t_i), t_i)$ is made amplitude independent by normalizing it by its maximum value over l.

2.2 Unknown item representation

In order to estimate tempo, beat and rhythm class for an unknown item u at frame t_i, we compute the set of $X_l(u, b_q, t_i)$ corresponding to a set of tempo hypothesis $b_q \in \{B\}$. If there are Q different tempo hypothesis, we compute Q different representations $X_l(u, b_q, t_i)$.

2.3 K-NN database construction

For a given audio item d, annotated into tempo over time $b_d(t_j)$, rhythm class c_d and beat positions $\{\tau\}_d$, we compute the corresponding complex vectors $X_l(d, b_d(t_j), t_j)$. For each frame j of each item d, we store the 48-dimensions vector $X_l(d, b_d(t_j), t_j)$ in the database and the corresponding annotated tempo $b_d(t_j)$, rhythm class c_d and the sub-set of normalized beat positions $\{\beta\}_{d,j}$.

Normalized beat positions $\{\beta\}_{d,j}$: If we note s_j and e_j the starting and ending time of frame t_j, the subset $\{\tau\}_{d,j}$ of beat positions are the τ_d for which $s_j \leq \tau_d$ and $\tau_d \leq e_j$. The normalized subset is then defined as $(\{\tau\}_{d,j} - s_j) \cdot b_d(t_j)$ and is noted $\{\beta\}_{d,j}$. It represents the beat markers of item d at frame j for a normalized beat frequency of 1Hz.

2.4 Search over the K-NN database

In order to estimate the parameters of a frame t_i of an unknown item u we perform a K-NN search. Since u is represented by Q complex vectors $X_l(u, b_q, t_i)$ (representing the Q tempo assumptions b_q), we perform Q searches. Given that X is a complex vector the search is performed using a distance in a complex space. For each of the Q complex vectors $X_l(u, b_q, t_i)$ representing a frame t_i of item u we compute its distance to each frame j of each item d of the database. The search is performed with K=1, i.e. we only consider the closest item of the K-NN search.

For the K-NN search, we tested several distances (we compare them in part 3):

Euclidean distance between the modulus of $X_l(u, b_q, t_i)$ and $X_l(d, b_d, t_j)$,

One-minus-cosine distance between the modulus of $X_l(u, b_q, t_i)$ and $X_l(d, b_d, t_j)$,

Complex distance between $X_l(u, b_q, t_i)$ and $X_l(d, b_d, t_j)$. The distance between the complex spectrum X_l and Y_l is defined by $d(X, Y, T) =$

$$\sum_l \sqrt{A_X^2(l) + A_Y^2(l) - 2A_X(l)A_Y(l)\cos(\Phi_X(l) - \Phi_Y(l, T))}$$

where A represents the modulus, Φ the phase, l the index in the complex vector, and T the best lag between the temporal signal $x(t)$ and $y(t)$ (corresponding to $X(l)$ and $Y(l)$) which minimizes the complex distance (maximizes the temporal synchronization). For this, each member T of a set of lags is tested and the phase spectrum of $Y(l)$ each time modified according to $\Phi_Y(l,T) = \Phi_Y(l) - 2\pi f_l T$. Because $X(u)/X(d)$ are independent of tempo, it is possible to compare two vectors with different initial tempo (this wouldn't be possible using the correlation between temporal sequences). Another advantage of a spectral computation of T (instead of a temporal correlation computation) is the possibility to give different weights to the various frequencies l in order to emphasize some of them.

2.5 Copy and scale the parameters

The result of the search therefore provides the reference to a frame j, item d, tempo assumption q and lag T which minimize the distance to (u, t_i): $(u,i) \rightarrow (d,j,q,T)$. **Rhythm class estimation:** From the Nearest Neighbor (j,d), we assign the rhythm class c_d to the unknown item at frame t_i. **Tempo estimation:** The tempo assigned to the unknown item at frame t_i is b_q (for the q minimizing the distance). **Beat-position estimation / de-normalized beat positions:** The beat positions assigned to the unknown item at frame t_i are given by $(\{\beta\}_{d,j} + T)/b_q$ for the q and T minimizing the distance.

2.6 Optimization

A set of optimization has been performed in order to reduce the computation time of the search.

1. For $\{B\}$ (the set of tempo-assumption for the unknown item), we only consider a subset of tempo. For this we perform a first tempo estimation, noted $\hat{b}(t_i)$, using the tempo estimation algorithm of [10], and we define, at each frame t_i, the set $\{B\}_i$ as the set of typical errors (1/3, 1/2, 1, 2, 3 the correct tempo) corresponding to $\hat{b}(t_i)$. $\{B\}_i$ is therefore defined as $[1/3, 1/2, 1, 2, 3]\hat{b}(t_i)$.

2. Only the K-NN database item (d,j) which initial tempo $b_d(j)$ is close to the candidate tempo b_q are considered. The closeness is defined as $\log_2(b_d(j)/b_q) < 0.3785$.

3. Since the computation time of the complex distance is high, the K-NN search is performed in two steps: 1) a rough search using a normal Euclidean distance or One-minus-cosine distance (therefore considering only the amplitude part of the DFT); 2) a fine search over the closest item using the complex distance to find the best alignment T between (u, b_q, i) and the tope ranked item (d,j).

Computation time: For the evaluation of part 3, the number of items of the 1-NN space is 10470. With the proposed optimization, the average number of searches in the database is 7491 (instead of (5*10470*0.9) and the average cost per unknown item of the process (search, complex distance, scaling) is 270ms using Matlab, an Intel Core 2 Duo 2.39GHz (only one processor used), 2Go of RAM.

3. EVALUATION

We evaluate the performances of the proposed approach for tempo estimation, rhythm classification and beat-tracking. We test the applicability of the method • when b = the ground-truth tempo for the computation of the unknown item's $X_l(u,b,t_i)$, • when b = the set of tempo assumption $\{B\}$ (the 5 tempo assumptions mentioned above) for the computation of it. We also compare the results obtained when using • the Euclidean distance (DE) and • the One-minus-cosine distance (DC). We finally compare the results obtained • at the frame level (the target is t_i), and • at the item level (the target is u). For the results at the item level, we have used a **late-fusion integration** method, i.e. the method is applied for all the frames t_i of u and a decision is taken from the whole set of frames of u. For this, a majority voting method is used for the classification and the median value over the frame's tempo is computed for tempo estimation. The "late-fusion integration" cannot be applied to the beat-tracking method. In all cases, we have used a Leave-One-Out evaluation method, i.e. testing in turn each frame t_i of each item u as a target, and removing each time all the frames belonging to u from the K-NN database.

		Known Tempo		Unknown Tempo	
		Frame-level	Item-level	Frame-level	Item-level
Class Acc.	DE	81.98	92.12	52.92	58.31
	DC	82.60	93.12	59.07	**65.76**
Tempo 4% (input tempo)	DE			60.35 (65.32)	61.46 (65.47)
	DC			66.79 (65.32)	**67.62 (65.47)**
Beat F-meas / Cemgil	DE	73.70 / 64.95		69.29 / 60.44	
	DC	74.00 / 65.19		**70.78 / 61.75**	

Tempo (filter class)	DE			98.24 (56.86)	97.54 (56.01)
	DC			97.88 (60.33)	95.86 (59.47)
Beat (filter class)	DE	75.76 / 67.18		82.71 / 73.85	
	DC	75.97 / 67.27		81.20 / 71.99	
Beat (filt. class tempo)	DE			83.59 / 74.72	
	DC			82.34 / 73.10	

Table 1: Performance measures for classification, tempo estimation and beat-tracking in the case of known and unknown tempo using distance DE and DC, frame-level and item-level decision.

3.1 Test-set

The evaluation is performed on the "ballroom dancer" test-set (as was used for the ISMIR2004 tempo induction contest) [7]. This test-set is often used for evaluation since it contains music for which the music genre and the rhythm class are closely related. It is composed of 698 tracks, each of 30 sec long, representing the following music genre: ChaCha (111 instances), Jive (60), QuickStep (82), Rumba (98), Samba (86), Tango (86), Viennese Waltz (65) and Slow Waltz (110). Annotations of beat positions have been made by the author and have been cross-checked several times.

3.2 Evaluation rules

The following rules are used for evaluation.

Classification: we have used the global accuracy (this is meaningful since the test-set is not highly unbalanced);

Tempo estimation: we have used the measure proposed by [7], i.e. we measure the number of tracks for which the estimated tempo is within a 4% Tolerance Window of the annotated tempo (without considering octave errors);

Beat-tracking: we have used the F-measure of Dixon [2] and the Gaussian error function of Cemgil [1].

3.3 Results and discussion

The results are shown in Table 1. Using **annotated tempo** ("Known tempo" column) for the creation of $X_l(u,b,t_i)$ and an Euclidean distance (DE) leads to 82% correct **class** recognition at the frame-level, 92% at the item-level. This accuracy slightly increases when using the One-minus-cosine distance (DC): 82.6% and 93.1%. As in many studies, we obtain better results using the One-minus-cosine distance than using the Euclidean distance.The **beat-tracking** performances (only applicable at the frame-level) are: 73.7% for the F-measure and 64.9% for Cemgil scores using DE. They also slightly increase when using DC: 74% and 65.2%. When considering only the frames for which the class has been correctly detected ("Beat (filter class)" row), i.e. the 82% remaining frames for DE and 82.6% for DC, the scores increases to 75.8% / 67.2% for DE and 76% / 67.3% for DC. Using **estimated tempo** ("Unknown tempo" column) for the creation of the $Q = 5$ versions of $X_l(u, b_q, t_i)$ and an Euclidean distance (DE) leads to 52.9% correct **class** recognition at the frame-level, 58.3% at the item-level. This accuracy largely increases when using the One-minus-cosine distance (DC): 59.1% and 65.8%. **Tempo** is correctly estimated at 60.4% at the frame-level and at 61.5% at the item-level using DE; at 66.8% at the frame-level and 67.6% at the item-level using DC. Remark that these results are above the ones obtained with the input tempo [10] (65.3%). The **beat-tracking performances** (only applicable at the frame-level) are: 69.3% F-measure and 60.4% for Cemgil scores using DE. They also slightly increase when using DC: 70.8% and 61.8%.

When considering only the tracks for which the class has been correctly identified ("Tempo (filter class)" row), i.e. the 59.07% remaining frames or the 65.76% remaining items, we obtain the following results for **Tempo** using DC: 97.9% at the frame-level 95.9% at the item level. It means that, if the class is correctly identified, the proposed approach succeeds to estimate the correct version of $X_l(u, b_q, t_i)$ among the $Q = 5$ versions in 98% of the cases. This very high recognition rate has to be compared with the one obtained on the same tracks (the ones correctly classified) by the algorithm used for estimating the input tempo [10]: 60.3% and 59.5%. We therefore think that improving the classification part of our approach could lead to a very good post-processing for tempo estimation algorithms. For the frames/ items for which the class has been correctly identified ("Beat (filter class)" row), the performance of the **beat-tracking** are 81.2% (F-measure) and 72% (Cemgil score) using DC.

We finally study the performance of the **beat-tracking** when considering only frames for which the class and the tempo have been correctly identified, i.e are within the 4% Tolerance Window ("Beat (filter class tempo)" row): 82.3% (F-measure) and 73.1% (Cemgil score) using DC. Given that the tempo accuracy is very high when the class is correct, adding a correct tempo filter makes little differences.

3.4 Comparison with previous results

Concerning **beat-tracking**, there is no previously published results on the "ballroom dancer" test-sets. Concerning **tempo estimation**, previous published results are in the ISMIR-2004 tempo induction contest [7]: 63.2% (excluding octave errors) / 92% (including octave errors) and in our paper [10]: 68.7% and 96.9%. The results obtained here (67.62% excluding octave errors) can therefore be considered as nearly equivalent with the ones obtained with dedicated signal processing algorithms. Concerning **classification** into rhythm classes, [3] obtained 85.7% track-based classification, we obtained 88% in [11]. Results obtained here (65.76%) with a 1-NN approach are therefore lower than the ones obtained in [11] with an AdaBoost classifier. Considering that the classification part our system can be improved (up to 88%) and considering that, for the part of correctly classified items our system reached 97.88% correct tempo estimation, one could therefore potentially reach a 87% correct tempo estimation (97.88% times 88%).

4. CONCLUSION AND FUTURE WORKS

In this paper, we proposed a new "copy and scale" method for estimating M.I.R. time-localized parameters. It relies on the use of complex spectrum as observation and a simple 1-NN with a distance in the complex domain. We apply this method for the case of beat-marker, tempo and rhythm classification. Using this direct approach on the "ballroom dancer" test-set, a classification accuracy of 65.8%, a tempo precision of 67.6% and a beat-tracking precision (F-measure) of 70.8% are obtained. Analysis of the results show that considering only the correctly classified frames leads to 97.9% tempo precision and 83.6% beat-tracking precision. The results presented here should only be considered as a proof of concept of our method. However, since the performances of each part of the proposed approach can easely be improved (using K-NN regression for tempo [4], sophisticated machine learning for rhythm classification [11] or introducing temporal continuity constraints in the decision), we believe this approach is promising.

Acknowledgments

Work supported by the French project Oseo "Quaero".

5. REFERENCES

[1] Cemgil, A., Kappen, B., Desain, P., and Honing, H. On tempo tracking: Tempogram representation and kalman filtering. *Journal of New Music Research 29*, 4 (2001), 259–273.

[2] Dixon, S. Evaluation of audio beat tracking system beatroot. *Journal of New Music Research 36*, 1 (2007), 39–51.

[3] Dixon, S., Gouyon, F., and Widmer, G. Towards characterisation of music via rhythmic patterns. In *ISMIR* (2004), pp. 509–516.

[4] Eronen, A., and Klapuri, A. Music tempo estimation with k-nn regression. *IEEE Trans on Audio, Speech and Language Processing 18*, 1 (2010), 50–57.

[5] Gouyon, F., and Dixon, S. A review of rhythm description systems. *Computer Music Journal 29*, 1 (2005), 34–54.

[6] Gouyon, F., Dixon, S., and Widmer, G. Evaluating low-level features for beat classification and tracking. In *IEEE ICASSP* (2007).

[7] Gouyon, F., Klapuri, A., and et al. An experimental comparison of audio tempo induction algorithms. *IEEE Trans. on ASLP 14*, 5 (2006), 1832–1844.

[8] Grosche, P., and Muller, M. A mid-level representation for capturing dominant tempo and pulse information in music recordings. In *ISMIR* (2009).

[9] MIREX. Audio beat tracking contest, 2009.

[10] Peeters, G. Template-based estimation of time-varying tempo. *EURASIP Journal on Applied Signal Processing 2007*, 1 (2007), 158–158.

[11] Peeters, G. Spectral and temporal periodicity representations of rhythm for the automatic classification of music audio signal. *submitted to IEEE. Trans. on ASLP* (2009).