Perceptual Tempo Estimation using GMM-Regression

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

Joachim Flocon-Cholet
STMP IRCAM-CNRS-UPMC
1, pl. Igor Stravinsky - 75004 Paris - France
Joachim.Flocon-Cholet@ircam.fr

ABSTRACT

Most current tempo estimation algorithms suffer from the so-called octave estimation problems (estimating twice, thrice, half or one-third of a reference tempo). However, it is difficult to qualify an error as octave error without a clear definition of what is the reference tempo. For this reason, and given that tempo is mostly a perceptual notion, we study here the estimation of perceptual tempo. We consider the perceptual tempo as defined by the results of the large-scale experiment made at Last-FM in 2011. We assume that the perception of tempo is related to the rate of variation of four musical attributes: the variation of energy, of harmonic changes, of spectral balance and short-term-event-repetitions. We then propose the use of GMM-Regression to find the relationship between the perceptual tempo and the four musical attributes. In an experiment, we show that the estimation of the tempo provided by GMM-Regression over these attributes outperforms the one provided by a state-of-the-art tempo estimation algorithm. For this task GMM-Regression also largely outperforms SVM-Regression. We finally study the estimation of three perceptual tempo classes ("Slow", "In Between", "Fast") using both GMM-Regression and SVM-Classification.

Categories and Subject Descriptors

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

Keywords

Perceptual Tempo, Tempo Class, GMM-Regression

1. INTRODUCTION

There has been and there is still many studies related to the estimation of tempo from an audio file (see [9] for a good overview). In the tempo estimation community it has been accepted that algorithms often make octave errors, i.e. estimating twice, thrice, half or on third of a "reference" tempo. This has led to the creation of two evaluation measures which consider either the estimation as correct if it is within 4% of the reference tempo (denoted by Accuracy-1), or also considering octave errors as correct (Accuracy-2).

Recently, focus has been made on trying to solve this octave-error problem, i.e. estimate exactly the "reference" tempo. This is partly motivated by the fact that many applications can simply not use a tempo estimation algorithm that produces octave errors (generating play-lists based on tempo continuity or searching music with slow-tempo are not possible in the presence of octave-errors). Studies on octave-error attempt to estimate exactly the "reference" tempo. But what is this "reference" tempo? Actually this "reference" tempo is often considered as a ground-truth, but in the case of tempo this is questionable considering that tempo is mainly a perceptual notion. For example, the experiment of Moelants and McKinney [12] highlighted the fact that people can perceived different tempi for a single track. Therefore, they propose to represent the tempo of a track as a histogram of its various perceived tempi[1]. Recently, Levy [11] did a large-scale experiment within the framework of Last-FM. In this experiment, 4000 tracks have been annotated using a crowd-sourcing method. Users were asked to select a speed for each track in a 3-point scale ("slow", "in between", "fast", "hard to say"), they were then asked to compare the track with a second track in terms of perception of speed, finally they were asked to tap along to provide a tempo estimation of the track.

1.1 Related works

Rather than a detailed overview of the works related to the octave-error problems or the estimation of perceptual tempo, we highlight here the differences between them.

The methods first differ by their **goals and methodologies**: estimation of perceptual tempo from scratch [18], estimation of an octave correction factor to be applied to a previous estimation made by a dedicated tempo estimation algorithm [19], estimation of tempo classes ("Slow", "Medium", "Fast" for [10]), use of the estimated tempo class to correct a previously estimated tempo [4], use of the class to create a prior to be used by a tempo estimation algorithm [8]. They then differ on the **audio features** chosen to represent the content of track: fluctuation patterns or autocorrelation function for [18], MFCCs for [19], vector representing the belonging to 101-moods for [4], large bag-of-features for [10], sophisticated harmonic and percussive rhythm features for [8]. They also differ on the **machine learning** method

[1]The current tempo-estimation measures of MIREX is actually derived from this.

MIRUM'12, November 2, 2012, Nara, Japan.

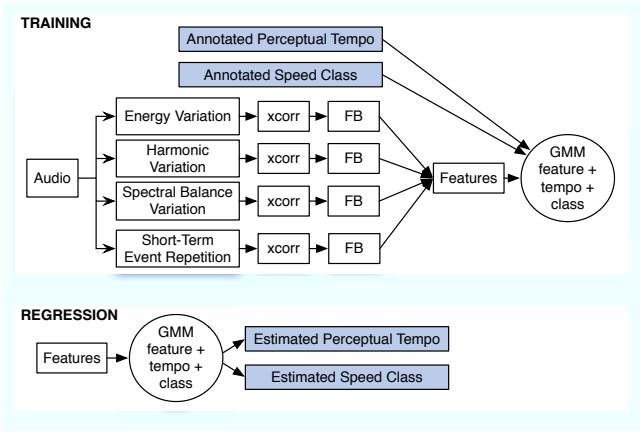

Figure 1: Overall schema of the proposed GMM training and GMM-Regression method

used: KNN for [18], GMM for [19], bag of classifiers (KNN, SVM, C4.5, AdaBoost ...) for [10], SVM for [4], SVM-Classification and SVM-Regression for [8]. They finally differ on the **data used**: "song" and "ballroom" test-set for [18, 19, 8], Last-FM user-tags and YouTube audio for [10].

1.2 Proposal and paper organization

While previous works rely mostly on energy-based features, timbre-features or a large bag-of-features, we start here from a set of assumptions related to the perception of tempo. For each of these assumptions we create a related audio feature. We assume that the perception of tempo is related to the rate of variation of four musical attributes: − the rate of variation of energy (as did the previous works) but also − the rate of variation of harmonic content, − the rate of variation of spectral balance (the distribution in high or low frequency of the energy) and − the rate of short-term-event-repetitions. We assume that a track with a rapid chord changes, rapid spectral-balance changes or rapid short-term repetitions will be perceived as fast even if the tempo of the sequencer was set to slow. The estimation of the related four feature-sets is explained in **part 2**.

We then create a model to find the relationship between the perceptual tempo, the perceptual tempo class and the four feature-sets. This model is then used to predict the perceptual tempo given the audio features. The model and the prediction is done using a technique borrowed from speech processing: GMM-Regression [5]. In [5], a GMM model is trained to learn the relationship between pitch and spectral envelope. The model is then used to predict the most-likely pitch given an observed spectral envelope. This is done using a regression over the values of the most-likely components of the GMM. The same method is applied here to predict the most-likely perceptual tempo and classes given the audio features. This is explained in **part 3**.

Surprisingly, most studies on octave-errors do not make use of a perceptual definition of the "reference" tempo. Their "reference" tempo (as the ones provided with the "song" or "ballroom" test-set) has often been only defined by one or two annotators. Therefore, it does not necessarily correspond to a shared perception of the tempo. As opposed to these studies, we rely here on the results of the large-scale experiment on perceptual tempo made by Last-FM [11]. From these results, we select only tracks for which the perception of perceptual tempo and perceptual tempo class is shared. On the resulting 1410 tracks test-set, we then measure the performances of GMM-Regression to predict the perceptual tempo and the perceptual tempo class. We also test the use of SVM-Classification and SVM-Regression. This is explained in **part 4**.

Finally we conclude in **part 5** and give directions for future works. The feature extraction, training and regression processes of our method are illustrated in Figure 1.

Difference with previous works: The closest works to our are the ones of [19] and [8]. [19] also tries to model the relationship between the features (MFCC in [19]) and the tempo using a GMM. However in [19] the GMM is only used to compute the most-likely combination of tempo+MFCC. In our case, the GMM is used to perform a regression, which provides directly the tempo. [8] also tries to perform regression but using SVM-Regression. However in [8], the regression is only used to estimate the class (the tempo is thereafter estimated using a peak-picking algorithm on the periodicity function). In our case, the GMM-Regression is used to provide directly the tempo.

2. AUDIO FEATURES

We explain here the four audio features related to our four assumptions concerning the variation of the content of the track. The extraction of the four feature sets is illustrated in Figure 2 on a real signal. In this figure, each row represents one of the feature-set.

2.1 Energy variation $f_{ener}(\tau)$

The variation of energy is represented by an onset-energy-function. We used the function we proposed in [15], named reassigned-spectral-energy-flux. We showed in [15] that this function allows to highlight onsets successfully even in case of weak onsets. Its computation is based on the time and frequency reassigned spectrum (in order to improve frequency separation and time location). The energy inside each frequency channel of this spectrum is converted to log-scale, low-pass filtered, differentiated over time and Half-Wave-Rectified (see [15] for more details). The final function is the sum over frequency of the individual functions. We denote it by $f_{ener}(t)$ where t denotes the time. In the following we consider as observation, the autocorrelation of this function denoted by $f_{ener}(\tau)$ where τ denotes "lags" in second. This is illustrated in the first row of Figure 2: column (a) represents the onset-energy-function and column (d) $f_{ener}(\tau)$.

2.2 Harmonic variation $f_{harmo}(\tau, T_z)$

Popular music is often based on a succession of harmonically homogeneous segments named "chords". The rate of this succession is proportional to the tempo (often one chord per bar). Rather than estimating the chord succession, we estimate the rate at which segments of stable harmonic content vary. For this we represent the harmonic content using chroma vectors using the method we proposed in [13]. In order to estimate the instant of changes between homogenous segments we use the "novelty score" proposed by [7]. This "novelty score" is obtained by convolving a Self-Similarity-Matrix (SSM) with a checkerboard kernel of size $2L$. The diagonal of the convolved matrix will have a large value at time t if the segments $[t-L, t]$ and $[t, t+L]$ are both homogenous but differ between each others. The diagonal therefore highlights instants where changes between stable parts occur. Since our assumption is that the rate of chord changes

is proportional to the tempo, L is chosen to be proportional to the tempo T_z ($L = 4 \cdot 60/T_z$). Since we do not have any prior on the tempo, we apply this method for various assumptions of tempo T_z. The resulting diagonals of the convolved matrices are then collected into a matrix denoted by $f_{harmo}(t, T_z)$. We then consider as observation the autocorrelation of each function, which we denote by $f_{harmo}(\tau, T_z)$ where τ denotes the "lags". This is illustrated in the second row of Figure 2: column (a) represents the SSM, column (b) $f_{harmo}(t, T_z)$ and column (c) $f_{harmo}(\tau, T_z)$.

2.3 Spectral balance variation $f_{specbal}(\tau, T_z)$

For music with drums, the balance between the energy content in high and low frequencies at a given time depends on the presence of the instruments: low > high if a kick is present, high > low when a snare is present. For a typical pop song in a 4/4 meter, we then observe over time t a variation of this balance at half the tempo rate. This variation can therefore be used to infer the tempo. In [17] we proposed a more robust method that compute, for a given tempo T_z, the likelihood that a given time t is a strong beat (1 or 3 in a 4/4 meter) or a weak beat (2 or 4). This is done by comparing the values of the balance function over a one bar duration. This feature is named spectral balance variation (see [17] for more details). Given that this function depends on a prior tempo T_z, we compute it for various tempo assumptions T_z in order to form a matrix, which we denote by $f_{specbal}(t, T_z)$. We then consider as observation the autocorrelation of each function, which we denote by $f_{specbal}(\tau, T_z)$ where τ denotes the "lags". This is illustrated in the third row of Figure 2: column (a) represents the spectrogram, column (b) $f_{specbal}(t_i, T_z)$ and column (c) $f_{specbal}(\tau, T_z)$..

2.4 Short-term event repetition $f_{repet}(\tau)$

We make the assumption that the perception of tempo is related to the rate of the short-term repetitions of events (such as the repetition of events with same pitch or same timbre). In order to highlight these repetitions, we compute a Self-Similarity-Matrix [6] (SSM) and measure the rate of repetitions in it. In order to represent the various type of repetitions (pitch or timbre repetitions) we use the method we proposed in [14]. We compute three SSMs corresponding to three different aspects of the content: the timbre (using MFCC features), the harmony (using chroma features) and the harmonic/noise content (using Spectral Crest/Valley features). We then compute a single SSM by summing the individual SSMs. The SSM $S(t_i, t_j)$ is a matrix where each entry represent the similarity between time t_i and time t_j. We convert it to a lag-matrix [1] $L(t_i, l_j)$ where $l_j = t_j - t_i$ denotes the "lag" between repetitions. In the lag-matrix, a high value in the column l_j indicates repetitions that occur systematically at a lag-interval of l_j. We then sum up the matrix over time t_i in order to obtain a vector representing the amount of repetitions at the various lags l. We denote this function by $f_{repet}(\tau)$ where τ denotes "lags". This is illustrated in the fourth row of Figure 2: column (a) represents the SSM and column (d) $f_{repet}(\tau)$.

2.5 Dimension reduction

The four feature sets are denoted by $f_{ener}(\tau)$, $f_{harmo}(\tau, T_z)$, $f_{specbal}(\tau, T_z)$, $f_{repet}(\tau)$ where τ denotes the lags (expressed in seconds) and T_z the various tempo assumptions. They are two possibilities to use these features to predict the tempo.

The first (which is partly used in [19]) is to (a) make a tempo assumption T_z, (b) use the column corresponding to T_z in $f_{harmo}(\tau, T_z)$, $f_{specbal}(\tau, T_z)$, (c) sample the four features sets $f_i(\tau)$ at lags τ corresponding to the sub-harmonics and harmonics of T_z, (d) measure the likelihood of the resulting combination of sampled-features and tempo assumption. However, this method was found very costly.

The second (which we use here) starts from the observation that the values of $f_{harmo}(\tau, T_z)$ and $f_{specbal}(\tau, T_z)$ do not depend too much on the tempo assumption T_z (see the example of part 2.6). They can therefore be reduced to $f_i(\tau)$ by summing their values over T_z. The four resulting vectors $f_i(\tau), i \in \{ener, harmo, specbal, repet\}$ still have a high dimensionality and are found too discriminative to be used for inferring tempi which are not exemplified in the training-set. We therefore applied a "blurring" technique over their lag-axis. This is done by applying a filter-bank (as [8] did) over their lag-axis. For this, we created 20 filters logarithmically spaced between 32 and 208bpm with a triangular shape. Each feature vectors $f_i(\tau)$ is then multiplied by this filter-bank leading to a 20-dim vector, denoted by $f_i(b)$ where $b \in [1, 20]$ denotes the filter.

To further reduce the dimensionality and de-correlated the various dimensions, we also tested the application of the Principal Component Analysis. Only the principal axis, which explain more than 10% of the overall variance, are kept. In the experiment of part 4, this usually leads to a reduction of 50% of the number of dimensions.

2.6 Illustrations

In Figure 2, we illustrate the computation of the four audio features (one on each row) on a real signal. As mentioned, the values of $f_{harmo}(\tau, T_z)$ and $f_{specbal}(\tau, T_z)$ do not depend too much on the tempo assumption T_z and are therefore summed up over T_z to obtain $f_{harmo}(\tau)$ and $f_{specbal}(\tau)$.

On column (d), we super-imposed to the plots the annotated tempo (continuous vertical line at 0.44s) as well as the various harmonics and sub-harmonics of the tempo (dashed lines). As can be seen on the top-part, the energy-variation function has a strong value at the eighth note (0.22s), the harmo-variation has two strong values around the whole-note (1.75s) indicating a sort of shuffle in the harmonic changes, the spectral-balance-variation has strong value at the half-note (0.875s), the short-term-event-repetition has a periodicity equal to the one of the tempo (0.44s). This figure illustrates a track with the following music model: fastest event (tatum) at the eight-note, chord variation once per bar (+shuffle), kick/ snare alternating twice per bar, event repetition at the quarter-note.

The functions $f_i(\tau)$ represented on the right-most column are then summarized using the filter-banks to create $f_i(b)$. The resulting functions $f_i(b)$ are illustrated in Figure 3. In this figure we represent $f_i(b)$ for all tracks of our test-set (see part 4). Each plot represents a specific feature i. The tracks have been sorted by increasing reference tempo. One can clearly see the specific patterns that expand while the tempo increases.

3. PREDICTION MODEL

For the prediction of the perceptual tempo T_e or the perceptual tempo class C_e we use the GMM-Regression prediction model proposed in [5]. We denote by $\boldsymbol{y}(l)$ the audio feature vector for track l (concatenation of the four feature-sets $f_i(b)$ for track l) and by x the parameter to be esti-

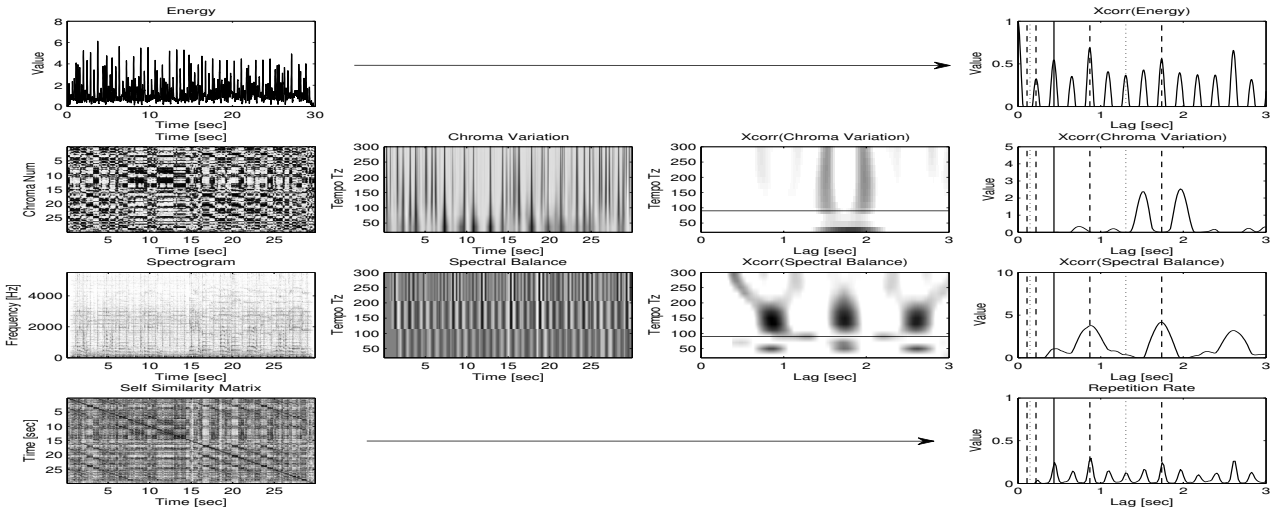

Figure 2: Example of feature extraction on audio signal: Big Audio Dynamite 'Looking For A Song" (7-Digital-ID: 4959116) (see part 2.6 for details).

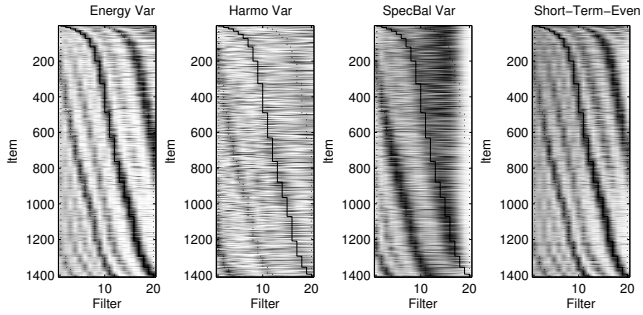

Figure 3: All feature-sets $f_{1,2,3,4}(b)$ for all tracks l of the test-set sorted by increasing (from top to bottom) annotated perceptual tempo $T_a(l)$.

mated. When we estimate the perceptual tempo of track l, we set $x(l) = T_e(l)$; when it is the perceptual tempo class, $x(l) = C_e(l)$. We then define $\boldsymbol{z}(l)$ to be the concatenation of $\boldsymbol{y}(l)$ and $x(l)$:

$$\boldsymbol{z}(l) = \begin{bmatrix} \boldsymbol{y}(l) \\ x(l) \end{bmatrix} \quad (1)$$

Given a set of tracks annotated into x (T_a or C_a) and the corresponding feature vectors $f_i(b)$ we train a Gaussian Mixture Model with K component and full-covariance-matrix using the Expectation Maximization algorithm. We denote by $\mathcal{N}$ the normal distribution, $\boldsymbol{\mu}_k$ the mean-vector of the component k and $\boldsymbol{\Sigma}_k$ its covariance matrix. We can subdivide $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ in the parts corresponding to $\boldsymbol{y}$ (denoted by $\boldsymbol{\mu}_k^y$ and $\boldsymbol{\Sigma}_k^{yy}$) and to x (denoted by $\boldsymbol{\mu}_k^x$ and $\boldsymbol{\Sigma}_k^{xx}$). The terms $\boldsymbol{\Sigma}_k^{xy}$ and $\boldsymbol{\Sigma}_k^{yx}$ represent the cross-dependency between x and $\boldsymbol{y}$ (hence between the parameter to be estimated and the audio features). For a given audio feature vector $\boldsymbol{y}$, x (its perceptual tempo T_e or its perceptual tempo class C_e) is then estimated in a maximum-likelihood way by

$$F(\boldsymbol{y}) = \mathbb{E}(x|\boldsymbol{y}) = \sum_{k=1}^{K} h_k \left[\boldsymbol{\mu}_k^x + \boldsymbol{\Sigma}_k^{xy}(\boldsymbol{\Sigma}_k^{yy})^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_k^y)\right] \quad (2)$$

with

$$h_k(\boldsymbol{y}) = \frac{\pi_k \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_k^y, \boldsymbol{\Sigma}_k^{yy})}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu}_k^y, \boldsymbol{\Sigma}_k^{yy})} \quad (3)$$

with

$$\boldsymbol{\mu}_k = \begin{bmatrix} \boldsymbol{\mu}_k^y \\ \boldsymbol{\mu}_k^x \end{bmatrix} \text{ and } \boldsymbol{\Sigma}_k = \begin{bmatrix} \boldsymbol{\Sigma}_k^{yy} & \boldsymbol{\Sigma}_k^{yx} \\ \boldsymbol{\Sigma}_k^{xy} & \boldsymbol{\Sigma}_k^{xx} \end{bmatrix} \quad (4)$$

4. EVALUATION

We evaluate here the performances of the four feature sets and the GMM-Regression to estimate the perceptual tempo and perceptual tempo classes.

4.1 Test-Set

The raw results corresponding to the Last-FM experiment are kindly provided by [11]. For each track l, it provides the whole set of annotations into **perceptual tempo (PT)** and into **perceptual tempo classes (PTC)**: "Slow", "In between", and "Fast". The test-set is made of 4006 items, which are provided without the audio. For 2425 items, at least 3 annotations are simultaneously provided into PT and PTC. For 1554 items, a majority[2] of annotators simultaneously agree on PT and PTC. We assigned to these items a "reference" PT and PTC, denoted by $T_a(l)$ and $C_a(l)$, defined as the median value of PT and PTC among the annotators that belong to the majority.

For each item, we used the 7-Digital API in order to access a 30s audio extract from which audio features has been extracted. This has been done querying the API using the provided artist, album and title names[3]. Matching have been found for 1410 items which is our final test-set. We provide the list of 7-Digital-ID used, the reference PT and PTC at the following URL: `http://recherche.ircam.fr/anasyn/peeters/pub/2012_ACMMIRUM/`.

[2]Majority is here defined as: at least 50% of the annotators agree (within 4%) on a reference tempo and class.

[3]When the API returned several items, only the first one was used. Due to this process, part of the audio tracks we used may not correspond to the ones used for the experiment of [11]. We estimate this part at 9%.

4.2 Measures

The performances are measured by comparing the estimation $T_e(l)$ and $C_e(l)$ to their references $T_a(l)$ and $C_a(l)$.

Class-Accuracy ($\mathbf{A_C}$): The quality of the class estimation is measured using class accuracy.

Tempo-Accuracy-8% ($\mathbf{A_T}$): Given that the manual annotations into $T_a(l)$ have been done using the spacebar of a keyboard (which we believe is not very accurate), given also that the filters used for dimensionality reduction do not allow a 4% precision in the full-range, we used an 8% relative precision, i.e. if $T_a(l)$=120bpm, we still consider as correct $T_e(l)$=110.4 or 129.6.

The evaluation has been performed using a ten-fold cross-validation, i.e. nine folds are used for training, the remaining one for testing. Each fold is successively used for testing; the results provided are average value over the ten-folds. Each fold has been created in order to guarantee the same tempo distribution and class-distribution as the original test-set.

4.3 Experimental protocol

In the following we test the estimation of the perceptual tempo T_e and class C_e using GMM-Regression. For this, the GMM is trained using the annotations (T_a and C_a) and the four feature-sets $f_i(b)$. Given that the range of the variables influences the creation of clusters in the GMM, we measure this by testing various scaling factors to be applied to T_a and C_a: $\alpha_T T_a$ and $\alpha_C C_a$. For the GMM, we test various numbers of components ($K = \{8, 16, 32, 64\}$).

T_e is then estimated directly as the output of eq. (2) using $x = T_e$. C_e is estimated in the same way using $x = C_e$. Given that the GMM-Regression provides a continuous value of C_e, we round it to its closest integer: C'_e = round$[C_e]$ (1="Low", 2="In Between", 3="Fast").

We test each features-set i separately as well as any combinations of them. We also test the influence of the dimension reduction by PCA. The implementation of the GMM-Regression is the one corresponding to [2], the implementation of the SVM-Classification and SVM-Regression is provided by the LibSVM library[3]. For SVM, we used a RBF kernel and an $\epsilon-$SVM for regression. Grid-search has been performed to find the best parameters (γ-parameter[4], cost-parameter and ϵ-parameters).

4.4 Results

We first create a set of base-lines:

Oracle: For the classification, we created a simple model which estimate $C_e(l)$ from the annotated $T_a(l)$. The evaluation is performed using a ten-fold cross-validation; each class is modeled using a single Gaussian model.

ircambeat : We then do the same using the tempo estimation $T_b(l)$ as provided by ircambeat [5] instead of the annotated one $T_a(l)$.

The results are indicated into Table 1 for both A_C and A_T. For each test, we performed an optimization over the parameters indicated in the "Configuration" column. Not indicated in the table is the fact that $K = 16$ was the best choice for all GMM tests.

Method	$\mathbf{A_C}$	$\mathbf{A_T}$	Configuration
Oracle	70	-	
ircambeat	51	67.3	
GMM-Regression			
1 (ener)	61.6		α_T=1, α_C=100, PCA=0
		67.9	α_T=1, α_C=1, PCA=0
2 (chroma)	45.1		α_T=0.005, α_C=1, PCA=0
		22.9	α_T=1, α_C=1, PCA=0
3 (specbal)	58.6		α_T=1, α_C=1, PCA=0
		51.3	α_T=1, α_C=1, PCA=0
4 (repet)	62.9		α_T=0.005, α_C=1, PCA=0
		66.8	α_T=1, α_C=1, PCA=0
1,2,3,4	61.3		α_T=1, α_C=100, PCA=1
		70.4	α_T=1, α_C=1, PCA=1
Best: 1,3,4	64.8		α_T=1, α_C=100, PCA=1
		72.9	α_T=1, α_C=100, PCA=0
SVM-Classification and SVM-Regression			
Best: 1,3,4	**68.3**		σ=1, C=31, PCA=1
		55.8	σ=1, C=171, ϵ=0.1, PCA=1

Table 1: Evaluation of Classification into Perceptual Tempo Classes (PTC) and Perceptual Tempo (PT) using various algorithms and configurations.

Tempo estimation: Among the four feature-sets (i=1,2,3,4), the best performing are the energy variation (i=1) and the short-term-event-repetition (i=4). The results obtained with only i=1 are very close to the ones obtained with ircambeat (A_T around 67%).

In the case of GMM and SVM, the best results for both A_C and A_T were obtained using a combination of the features related to energy variation (i=1), spectral-balance variation (i=3) and short-term-event-repetitions (i=4). Therefore without the use of the harmo-variation (i=2). Actually the pattern-over-tempo of the harmo-variation in Figure 3 is also more fuzzy than the ones of the other features. Given the high dimensionality resulting for the combination of the three feature-sets (60 dim), the use of the PCA was found useful in most cases. Also, in the case of GMM, increasing α_C to 100 (in order to favor the creation of clusters representing classes in the GMM) was found beneficial.

On overall, GMM-Regression provides the best results for tempo estimation (A_T=72.9%). It provides a tempo estimation, which is 5.6% higher than ircambeat . The results obtained with SVM-Regression (A_T=55.8%) are largely below the ones obtained with GMM-Regression. In Figure 4 we present details of the tempo-estimation obtained using ircambeat and the best GMM-Regression configuration. The three corridors (indicated by dotted-lines) correspond to the 8% relative precision for the exact tempo, half and twice of it. According to these two figures, we see that the amount of doubling octave errors produced by the GMM-Regression is much lower (5% against 20%) but the halving octave errors are larger (9% against 3%).

As comparison, we provide the results published in [11] on the tempo-accuracy obtained using the EchoNest algorithm (40.7%), BpmList (69.1%), Vamp (58.3%). It should be noted however that the methods used in [11] to select tracks, to infer the reference TP and to compute the Tempo-Accuracy differ from the present ones chosen here.

[4]In practice, we optimize the parameters σ which is independent of the dimensionality: $\gamma = 1/(D \cdot \sigma^2)$ where D is the number of dimensions.

[5]ircambeat can be considered as a good representation of current state-of-the-art algorithm. It ranked first in the MIREX-2005 "At-Least-One-Tempo-Correct" estimation task and currently perform among the best in the MIREX beat-tracking task.

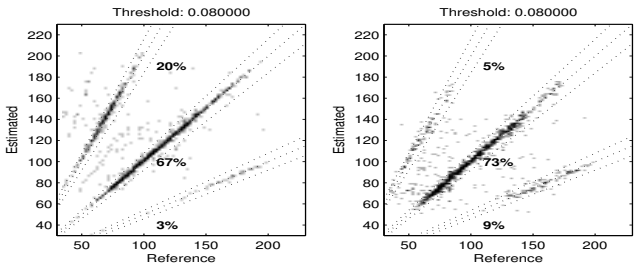

Figure 4: Detailed results of tempo estimation using [LEFT] ircambeat [RIGHT] GMM-Regression.

Class estimation: SVM-Classification provides the best results (A_C=68.3%). It actually provides class estimation (A_C=68.3%) very close to the one obtained using the Oracle based on annotated tempo (A_C=70%). Since our best tempo estimation is largely below 100%, this means that our feature-sets are able to catch characteristics in the audio that are related to the perception of the tempo classes but which are not useful for tempo estimation.

"Ballroom test-set": Finally, for comparison, we indicate in Table 2 the results obtained on the 'ballroom" test-set. The best results are obtained using GMM-Reg: A_T=87% (ircambeat achieves 66.1%). As comparison, the best results so far were the ones of [18] with 78.51% (but with a more restrictive 4% accuracy). For this test-set, SVM achieves A_C = 88.5% which is slightly higher than the 88% we obtained in [16] with our previous classification method.

Method	$\mathbf{A_C}$	$\mathbf{A_T}$
ircambeat		66.1
GMM-Regression	80.37	**87**
SVM-Classification and Regression	**88.5**	55.1

Table 2: Evaluation on the "ballroom" test-set.

5. CONCLUSION AND FUTURE WORKS

In this paper, we studied the estimation of perceptual tempo (as defined by the results of the large-scale experiment made at Last-FM) using four assumptions related to the rate of variations of musical attributes and a prediction using GMM-Regression. We showed that using three of these assumptions (rate of variation of the energy, of the harmonic changes and short-term-event-repetitions) allows to estimate the perceptual tempo at 73%, i.e. better than using a state-of-the-art tempo estimation algorithm. We also showed that, for this task, GMM-Regression largely outperforms SVM-Regression. For classification into perceptual tempo classes, we showed that SVM-Classification outperforms GMM-Regression. It allows achieving results (68.3%) very close to the ones obtained by an Oracle knowing the annotated tempo.

The use of the fourth assumption (rate of harmonic changes) was not successful in our experiment. This may be due to the fact that harmonic changes are not enough periodic to be model as a periodic signal, or may be due to the estimator we used to measure the harmonic changes. Further works will concentrate on improving this estimator. Further works will also concentrate on using the full range of annotations provided for each track rather than the single majority perceptual tempo derived from it.

Acknowledgments

This work was partly supported by the Quaero Program funded by Oseo French agency and by the MIReS project funded by EU-FP7-ICT-2011.1.5-287711.

6. REFERENCES

[1] M. Bartsch and G. Wakefield. To catch a chorus: Using chroma-based representations for audio thumbnailing. In *Proc. of IEEE WASPAA*, New Paltz, NY, USA, 2001.

[2] S. Calinon, F. Guenter, and A. Billard. On learning, representing and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 37(2):286–298, 2007.

[3] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Trans. on Intelligent Systems and Technology*, 2(27):1–27, 2011.

[4] C. Chen, M. Cremer, K. Lee, P. DiMaria, and H. Wu. Improving perceived tempo estimation by statistical modeling of higher level musical descriptors. In *Proc. of the 126th AES Convention*, 2009.

[5] T. En-Najjary, O. Rosec, and T. Chonavel. A new method for pitch prediction from spectral envelope and its application in voice conversion. In *Proc. of Eurospeech*, Geneva, 2003.

[6] J. Foote. Visualizing music and audio using self-similarity. In *Proc. of ACM Int. Conf. on Multimedia*, pages 77–80, Orlando, Florida, USA, 1999.

[7] J. Foote. Automatic audio segmentation using a measure of audio novelty. In *Proc. of IEEE ICME*, New York City, NY, USA, 2000.

[8] A. Gkiokas, V. Katsouros, and G. Carayannis. Reducing tempo octave errors by periodicity vector coding and svm learning. In *Proc. of ISMIR*, Porto, Portugal, 2012.

[9] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Trans. on Audio, Speech and Language Processing*, 14(5):1832–1844, 2006.

[10] J. Hockman and I. Fujinaga. Fast vs slow: Learning tempo octaves from user data. In *Proc. of ISMIR*, Ultrecht, The Nederlands, 2010.

[11] M. Levy. Improving perceptual tempo estimation with crowd-sourced annotations. In *Proc. of ISMIR*, Miami, USA, 2011.

[12] D. Moelants and M. F. McKinney. Tempo perception and musical content: What makes a piece slow, fast, or temporally ambiguous? In *Proc. of ICMPC*. Evanston, IL, 2004.

[13] G. Peeters. Chroma-based estimation of musical key from audio-signal analysis. In *Proc. of ISMIR*, pages 115–120, Victoria, Canada, 2006.

[14] G. Peeters. Sequence representation of music structure using higher-order similarity matrix and maximum-likelihood approach. In *Proc. of ISMIR*, Vienna, Austria, 2007.

[15] G. Peeters. Template-based estimation of time-varying tempo. *EURASIP Journal on Applied Signal Processing*, 2007(1):158–158, 2007. doi:10.1155/2007/67215.

[16] G. Peeters. Spectral and temporal periodicity representations of rhythm for the automatic classification of music audio signal. *IEEE Trans. on Audio, Speech and Language Processing*, 19(5):1242–1252, July 2011.

[17] G. Peeters and H. Papadopoulos. Simultaneous beat and downbeat-tracking using a probabilistic framework: theory and large-scale evaluation. *IEEE Trans. on Audio, Speech and Language Processing*, 19(6):1754–1769, August 2011.

[18] K. Seyerlehner, G. Widmer, and D. Schnitzer. From rhythm patterns to perceived tempo. In *Proc. of ISMIR*, 2007.

[19] L. Xiao, A. Tian, W. Li, and J. Zhou. Using a stastic model to capture the association between timbre and perceived tempo. In *Proc. of ISMIR*, Philadeplhia, USA, 2008.