Journal of New Music Research
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/nnmr20

Melody Harmonization With Interpolated Probabilistic Models

Stanisław A. Raczyński[a], Satoru Fukayama[b] & Emmanuel Vincent[c]
[a] Gdansk University of Technology, Poland
[b] The University of Tokyo, Japan
[c] INRIA, France
Published online: 22 Oct 2013.

To cite this article: Stanisław A. Raczyński, Satoru Fukayama & Emmanuel Vincent (2013) Melody Harmonization With Interpolated Probabilistic Models, Journal of New Music Research, 42:3, 223-235, DOI: 10.1080/09298215.2013.822000

To link to this article: http://dx.doi.org/10.1080/09298215.2013.822000

Journal of New Music Research, 2013
Vol. 42, No. 3, 223–235, http://dx.doi.org/10.1080/09298215.2013.822000

Melody Harmonization With Interpolated Probabilistic Models

Stanisław A. Raczyński[1], Satoru Fukayama[2] and Emmanuel Vincent[3]

[1]*Gdansk University of Technology, Poland;* [2]*The University of Tokyo, Japan;* [3]*INRIA, France*

Abstract

Most melody harmonization systems use the *generative* hidden Markov model (HMM), which model the relation between the hidden chords and the observed melody. Relations to other variables, such as the tonality or the metric structure, are handled by training multiple HMMs or are ignored. In this paper, we propose a *discriminative* means of combining multiple probabilistic models of various musical variables by means of model interpolation. We evaluate our models in terms of their cross-entropy and their performance in harmonization experiments. The proposed model offered higher chord root accuracy than the reference musicological rule-based harmonizer by up to 5% absolute.

1. Introduction

Automatic melody harmonization is the process of determining the most musically suitable chordal accompaniment to a given monophonic melody. It is an important part of musical composition and so it is a common exercise in all music composition classes, where it typically involves determining a *four-part harmony*, i.e. the movement of four voices, namely: soprano, alto, tenor and bass. The task can be either to find the harmonization for a given melody performed by the soprano voice, or to find the three other voices for a given bass line (*unfigured bass*), often with some chordal information given (*figured bass*). In this work, however, we focus on harmonization in the more narrow sense of generating a sequence of background chords matching a given melody (Gang, Lehman, & Wagner, 1998), which can be played on supporting instruments, e.g. on a guitar or a piano. It is simpler than four-part harmonization because one does not need to determine the exact movements of the voices—it does not include inversions, added and removed tones, etc., but only the root pitch (C, C♯, etc.) and the chord type (major, minor, etc.). The melody together with the chord labels are typically referred to as *lead sheets*, an example of which is shown in Figure 1.

Harmonization is a necessary step in most algorithmic composition methods, which typically involve generating a melodic line first, and afterwards supplementing it with an accompaniment. For example, in their *Orpheus* automatic composition system, Fukayama, Nakatsuma, Sako, Nishimoto, and Sagayama (2010) compose a melody based on constraints resulting from the tonal accent in the lyrics and from basic musicological rules, and then compose the chordal accompaniment using a collection of accompaniment patterns. Harmonization has also been explored as an easy way to create polyphonic ring-tones from simple melodies in the *i-Ring* ring-tone harmonization system by Lee and Jang (2004). Furthermore, automatic harmonization has recently received significant commercial interest as an easy way for non-musicians to create well-sounding music based on simple melodies. The most well-known implementations are the *MySong* software developed in cooperation with Microsoft (Simon, Morris, & Basu, 2008) and the commercial software package *Band-in-a-Box* (BIAB) (PG Music Inc., 2012). Both are designed as tools for non-professional musicians, or even non-musicians, to create songs with instrumental accompaniment by singing a melody into a microphone. All of the above methods perform a lead sheet-like harmonization.

The history of automatic harmonization is much older, however. Some of the earliest attempts at harmonization were made by Winograd (1968), and Rothgeb (1669, 1979). Steels (1986) proposed a heuristic search approach, and by Ebcioğlu (1986, 1988), who proposed a rule-based system targeting Bach's chorales. Another rule-based harmonizer based on musicological expertise—called *Harmonic Analyzer*—was more recently developed by Temperley and Sleator (1999, 2012). Other harmonization methods have also been explored. Phon-Amnuaisuk and Wiggins (1999) developed a prototypesystem

Correspondence: Stanisław A. Raczyński, Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, ul. G. Narutowicza 11/12, 80-233 Gdańsk-Wrzeszcz, Poland. E-mail: staraczy@pg.gda.pl

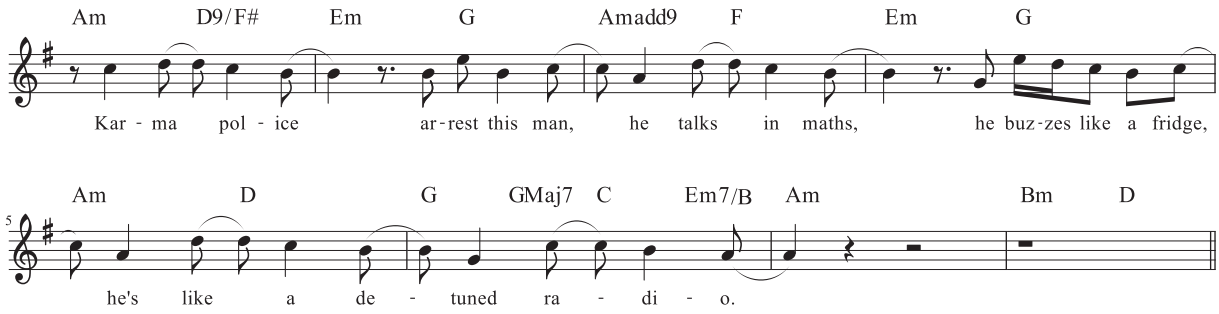

Fig. 1. Example harmonization of the lead melody from Radiohead's *Karma Police*.

based on genetic algorithms with knowledge-rich structures. Constraint satisfaction systems were investigated by Pachet and Roy (2001) and sequential neural networks by Gang et al. (1998). Neural networks were also used by Cunha and Ramalho (1999) in their hybrid neural-musicological real-time chord prediction system. Another hybrid harmonization system that combines Hidden Markov Models (HMMs) with a set of heuristic rules for rapid training was proposed by Chuan (2007). Among these, only the systems of Temperley and Sleator (2012), Gang et al. (1998), Cunha and Ramalho (1999), and Chuan and Chew (2007) are lead sheet-like harmonization systems.

Among the most flexible systems are those based on unsupervised probabilistic modelling, which typically utilize HMMs. Single-HMM approaches include the *MySong* software and an implementation based on *MySong* from Chuan (2011), as well as the *i-Ring* ring-tone harmonization system by Lee and Jang (2004). A slightly more complicated four-part harmonization approach using a dual HMM (one for harmonization and another one for ornamentation) was proposed by Allan and Williams (2005). Later, Paiement, Eck, and Bengio (2006) proposed a very sophisticated, multi-level graphical model for modelling chord movement with respect to the melody, which is capable of modelling long-term relationships between chords, as opposed to HMMs that are only capable of modelling short-term dependencies. However, being a non-dynamic graphical network, their model is limited to fixed-length songs (of exactly 16 bars). By contrast with musicological rule-based methods, which require careful formulation and application of harmonization rules for particular genres (e.g. classical or jazz), these probabilistic methods aim to automatically infer those rules from a corpus of example data and are therefore applicable to all genres, even when musicological expertise is not available.

All of the above probabilistic harmonization systems are however limited to modelling the relation between a single hidden layer (chord sequence) and a single observed layer (melody), without any explicit mechanism to include models of other relevant musical quantities, such as the key and the current tonal centre, the rhythm and the musical accent, or the genre, period and the composer. One can use multiple HMMs corresponding to, e.g. different genres or keys as in Simon (2008), but with many variables this approach quickly suffers from over-fitting. In this paper we propose to build versatile chord models by interpolating between a collection of simpler *sub-models* using linear or log-linear interpolation.

This paper is organized as follows. Section 2 explains the proposed modelling and training approach and Section 3 gives details about the particular sub-models used in our experiments. The experimental set-up and the results are described in Section 4. Finally, the conclusion is given in Section 5.

2. General approach

When figuring out the accompaniment, one needs to keep in mind the basic rules of tonal music: the tonality (everything in tonal music happens with respect to the tonic), the chord progressions (certain chord progressions are more natural and pleasant, e.g. progressions corresponding to the circle of fifths, progressions descending by thirds and common cadences), and of course harmonic compatibility with the melody. In the *generative*, HMM-based systems, the current chord is the underlying state C_t, while the observation is the melody M_t. Chord progression is modelled with a Markov chain $\mathrm{P}(C_t|C_{t-1})$ and the observed melody by a multinomial distribution conditioned on the system's state $\mathrm{P}(M_t|C_t)$ (see the top of Figure 2).

2.1 Model structure

We propose a more flexible way of developing probabilistic harmonization models, in which the time-varying tonality T_t, as well as other musical variables can be explicitly taken into account. We propose a *discriminative* model, in which the chords are modelled conditionally on all other variables (see bottom of Figure 2):

$$\mathrm{P}(C_t|\mathbf{C}_{1:t-1}, \mathbf{X}_{1:t}), \tag{1}$$

where $1:t$ denotes a range of time indices from the first to the current time frame and $\mathbf{X}$ is a set of other musical variables, such as the melody, the tonal centre, the metrical accent, the style or genre, etc. Discriminative models have been found to outperform generative models in many fields (Jebara, 2004), such as speech recognition (Rathinavelu & Deng, 1996; Woodland & Povey, 2002), machine translation

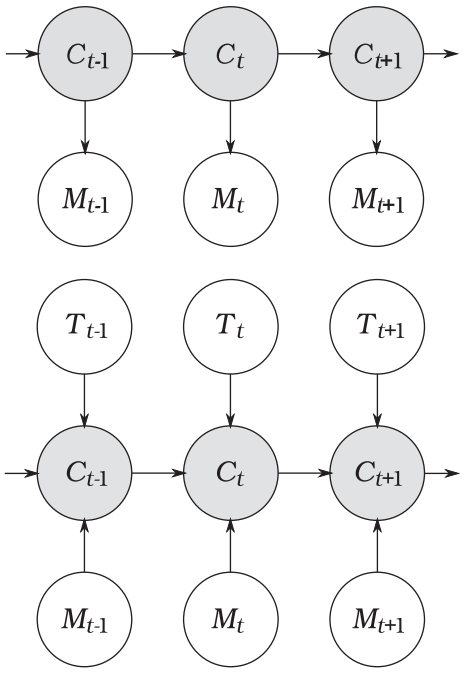

Fig. 2. A typical HMM for melody harmonization (top) compared to the proposed model (bottom).

(Och, 2002), text classification (Rennie, 2001) or genomics (Jaakkola, Diekhans, & Haussler, 2000).

This conditional multinomial distribution has too many parameters to be used in practice, hence we approximate it by interpolating between multiple sub-models P_i involving a different subset of conditioning variables $\mathbf{A}_{i,t} \subset \{C_{1:t-1}, \mathbf{X}_{1:t}\}$. The interpolation can be linear,

$$\mathrm{P}(C_t|\mathbf{C}_{1:t-1}, \mathbf{X}_{1:t}) = \sum_{i=1}^{I} a_i \mathrm{P}_i(C_t|\mathbf{A}_{i,t}), \tag{2}$$

with

$$\sum_{i=0}^{I} a_i = 1, \tag{3}$$

or log-linear,

$$\mathrm{P}(C_t|\mathbf{C}_{1:t-1}, \mathbf{X}_{1:t}) = Z^{-1} \prod_{i=1}^{I} \mathrm{P}_i(C_t|\mathbf{A}_{i,t})^{b_i}, \tag{4}$$

where I is the number of sub-models, $a_i \geq 0$ and $b_i \geq 0$ for $i = 1, \ldots, I$ are the interpolation coefficients and

$$Z = \sum_{C_t} \prod_{i=1}^{I} \mathrm{P}_i(C_t|\mathbf{A}_{i,t})^{b_i} \tag{5}$$

is a normalizing factor depending on $\mathbf{A}_{i,t}$. For example, we will consider in the following: $\mathbf{A}_{1,t} = \{C_{t-1}\}$, $\mathbf{A}_{2,t} = \{T_t\}$ and $\mathbf{A}_{3,t} = \{M_t\}$.

Linear (Jelinek & Mercer, 1980) and log-linear (Klakow, 1998) interpolation have been previously used in the context of natural (spoken) language modelling to combine models with different temporal spans (n-grams with different values of n). Here we have generalized this approach to interpolate between sub-models conditioned on different musical variables.

2.2 Smoothing

Although the sub-models P_i now have fewer parameters, over-fitting issues may still arise due to data sparsity, so the above equations are not directly usable. In order to address these issues, each of the sub-models must be smoothed (Zhai & Lafferty, 2004). In this study we perform smoothing by combining each sub-model with the prior chord distribution and a uniform distribution. In the case of linear interpolation, the smoothing is applied to the interpolated model:

$$\mathrm{P}(C_t|\mathbf{C}_{1:t-1}, \mathbf{X}_{1:t}) = \alpha \mathrm{P}(C_t) + \beta + \sum_{i=1}^{I} a_i \mathrm{P}_i(C_t|\mathbf{A}_{i,t}), \tag{6}$$

with

$$\alpha + \beta + \sum_{i=1}^{I} a_i = 1. \tag{7}$$

In the case of the log-linear interpolation, each model is smoothed separately before combining them:

$$\mathrm{P}(C_t|\mathbf{C}_{1:t-1}, \mathbf{X}_{1:t}) = Z^{-1} \prod_{i=1}^{I} \left(\gamma_i \mathrm{P}_i(C_t|\mathbf{A}_{i,t}) + \delta_i \mathrm{P}(C_t) + \epsilon_i\right)^{b_i}, \tag{8}$$

with

$$\gamma_i + \delta_i + \epsilon_i = 1 \tag{9}$$

for all i and

$$Z = \sum_{C_t} \prod_{i=1}^{I} \left(\gamma_i \mathrm{P}_i(C_t|\mathbf{A}_{i,t}) + \delta_i \mathrm{P}(C_t) + \epsilon_i\right)^{b_i}. \tag{10}$$

2.3 Training

The proposed models are trained on two disjoint sets of example data called *training set* and *validation set*. The sub-models P_i and the prior distribution $\mathrm{P}(C_t)$ are first trained in the maximum likelihood (ML) sense on the training set by counting occurrences (Zhai & Lafferty, 2004). The interpolation coefficients a_i or b_i and the smoothing coefficients α and β or γ_i, δ_i and ϵ_i are then jointly trained on the validation set according to one of two possible training objectives.

Classical generative training is achieved by estimating the interpolation and smoothing coefficients in the ML sense on the validation set. Because the log-likelihood is convex (Klakow, 1998), any optimization algorithm can be used. In the following, we have used a non-negatively constrained limited-memory Broyden–Fletcher–Goldfarb–Shanno (BFGS) method (a quasi-Newton optimization), built into the GNU R environment as the `optim()` function (R Development Core Team, 2011).

Even though the likelihood is a common evaluation measure for statistical models, higher likelihood does not always

translate into better performance for the considered application. For example, it is well known in the field of automatic speech recognition that the likelihood of language models can sometimes be improved without effect on the word error rate (Chen, Beeferman, & Rosenfeld, 1998). For that reason, we alternatively propose to perform discriminative training by estimating the interpolation and smoothing coefficients so as to maximize the chord root note accuracy on the validation set, which is the main evaluation metric used in the harmonization experiments in Subsection 4.2. Because this metric is not differentiable, gradient-based methods cannot be used and we have applied the following multi-step brute-force search. First, all smoothing coefficients were fixed to $\alpha_i = 0.1$ and $\beta_i = 0.5$ (values chosen experimentally) and the interpolation coefficients were optimized by testing all combinations of values between 0 and 1 in 0.1 steps (11 distinct values). Then, the smoothing coefficient pairs were optimized separately and sequentially in the same range. Finally, the interpolation coefficients were fine-tuned around the original optimum (±20%, 11 values) using the newly trained smoothing coefficients.

3. Sub-models

As a proof of concept, we have developed three sub-models that model the three most important aspects of chords: chord progressions, relation to the tonality and relation to the melody:

$$\mathrm{P}_1 = \mathrm{P}(C_t|C_{t-1}), \tag{11}$$

$$\mathrm{P}_2 = \mathrm{P}(C_t|T_t), \tag{12}$$

$$\mathrm{P}_3 = \mathrm{P}(C_t|M_t). \tag{13}$$

To train these three sub-models, we have used a collection of around 2000 lead sheets (melodies annotated with keys and absolute chord labels) encoded in the MusicXML format that are freely available on the Wikifonia web page (Wikifonia Foundation, 2012), because it is the largest existing collection of lead sheets. This included mostly popular (e.g. pop, rock) songs from the twentieth and the twenty-first centuries. Due to user-contributed nature of the data, the songs were first screened for improper chord labels and wrong keys. It is possible that a small proportion of data remained inaccurate after this screening process, however this may have had only a minor impact on the results (typically, a small decrease of the absolute performance of the tested algorithms) due to the much larger proportion of correct data. We converted each lead sheet into three sequences of symbols representing tonality, melody and chords. In order to do so, we first partitioned each input into regular time frames of length 1/3, 1/2, 1, 2, 4, 8 or 16 beats. For every frame, we defined the melody variable M_t as the unordered list of pitch classes of the 12-tone chromatic scale appearing in the melody, which can take $2^{12} = 4096$ distinct values. This is similar to the melody encoding in Chuan (2011), and Raphael and Stoddard (2004). The tonality T_t was encoded as one of 24 different key labels resulting from the combination of 12 tonics (C, C♯, D, D♯, E, F, F♯, G, G♯, A, A♯, B) and 2 modes (major or minor). The chord C_t was labelled by one of 13 root pitch classes (C, C♯, D, D♯, E, F, F♯, G, G♯, A, A♯, B or 'none' for non-chords) and one of 27 chord types (major, minor, dominant, diminished, half-diminished, augmented, power, suspended-second, suspended-fourth, major-sixth, minor-sixth, major-seventh, minor-seventh, dominant-seventh, diminished-seventh, augmented-seventh, major-ninth, minor-ninth, dominant-ninth, augmented-ninth, minor-eleventh, dominant-eleventh, major-minor, minor-major, major-thirteenth, dominant-thirteenth or 'none' for non-chords), resulting in $N = 351$ distinct chord labels in total. The chord C_0 before the beginning of the song was assumed by convention to be 'none'. In the case of a time frame containing more than one key or chord, the longest lasting key and chord labels within that frame were selected.

The distribution of tonalities in the training set is shown in Figure 3. The C-major key appears to be dominant, which will have an impact on the design of the models in order not to bias them toward that particular tonality, as explained in Subsection 3.2.

3.1 Chord prior

The chord prior $\mathrm{P}(C_t)$ is used for smoothing and as a reference model in the evaluation. The prior chord distribution trained on the training dataset is presented in Figure 4. Note that the major, dominant, minor, minor- and major-seventh chords make up the vast majority of the chords in the dataset. Also, due to the dominance of the C-major key in the training set, the pitch classes C, F and G have visibly higher probabilities than the other root pitch classes.

3.2 Chord bigram model

The chord progression model is built under the Markov assumption, following the HMM-based approaches of Chuan (2011), Lee and Jang (2004), and Allan and Williams (2005), resulting in a bigram model $\mathrm{P}_1(C_t|C_{t-1})$. Longer-term dependencies have been studied by Paiement et al. (2006), and Scholz, Vincent and Bimbot (2009). Meredith (2007, pp. 326–328) suggested that the 'post-context', i.e. the future notes and chords, can carry valuable information about the past. These sub-models could eventually be used in our framework but they are beyond the scope of the current paper, which is to provide a proof of concept of model interpolation in MIR.

In order to avoid problems with data sparsity, the model was trained with state tying: probabilities of all *relative* chord transitions were tied together, so for example the probability of transition from C-major to G-minor (seven semitones, I–V transition in C-major) is identical to that of transition from G-major to D-minor (also seven semitones, I–V transition in G-major). This is motivated by the observation that in tonal music songs can be freely transposed between all keys without any loss of musical correctness (Papadopoulos & Peeters, 2007) and that the relative chord transitions should have the same probabilities in different keys. The chord prior did not

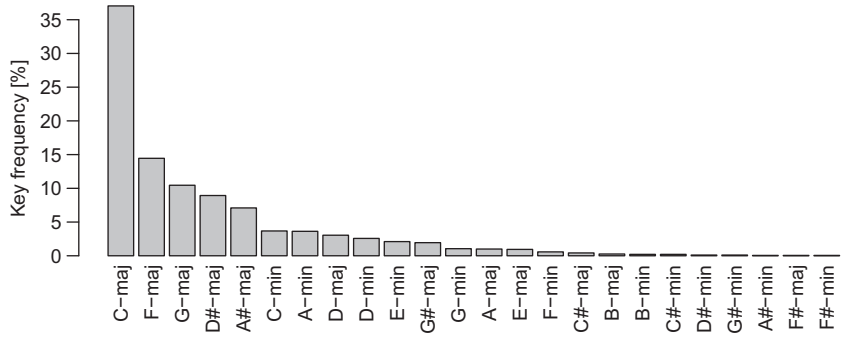

Fig. 3. Histogram of the key labels in the training dataset.

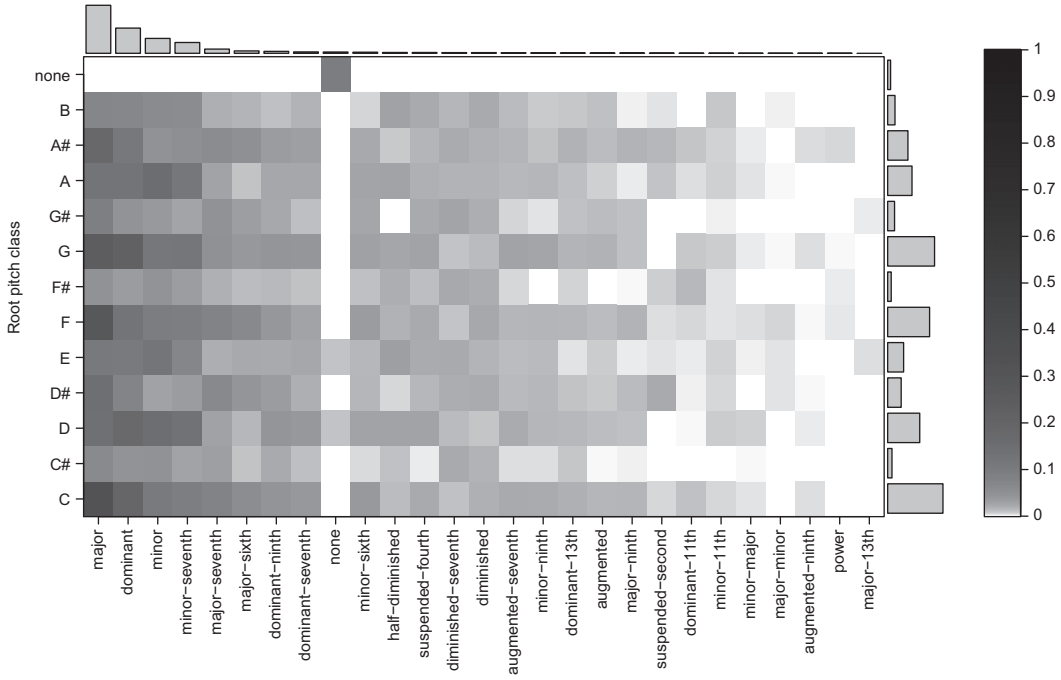

Fig. 4. Distribution of chords $P(C_t)$ for 1-beat frames. On top: distribution of chord types; on the right: distribution of chord roots. Chord types are sorted by their occurrence frequency.

include state tying, because it requires much fewer parameters, and therefore the bias is not that strong, and because it does not aim at modelling relative chord transitions.

The resulting conditional chord distribution can be observed in Figure 5. The distribution for a 1-beat analysis frame (top of Figure 5) is very concentrated towards the previous chord (G-major to G-major transition), because chords typically last for at least a few beats. On the other hand, using a 16-beat analysis frame makes the bigram probabilities more evenly distributed, though still dominated by the transition to the same chord.

3.3 Tonality model

In the tonality model $P_2(C_t|T_t)$, for the same reasons as explained in Subsection 3.2, the chords corresponding to the same scale degree in different keys were tied together. In other words, observing, e.g. a dominant major chord in one key increases the probability of dominant major chords in all keys. The resulting tonality model distribution is depicted in Figure 6. Notice that knowing the current tonality to be, in this case, $T_t =$ C-major increases the dominance of the most common degrees: the dominant (V), subdominant (IV), supertonic (ii), but mostly the tonic (I).

3.4 Melody model

Finally, for the same reasons again, state tying was used for the melody model $P_3(C_t|M_t)$ as well. Note patterns with the same content relative to the chord root were given identical probabilities, e.g. the unordered note combination (C,G) in the chord of C-major is equally probable as the note combination (D♯,A♯) in the chord of D♯-major. The resulting melody model distribution is shown in Figure 7. Note that having more melodic information, i.e. more notes in a frame (here C, E and G in one frame in the bottom part of the plot),

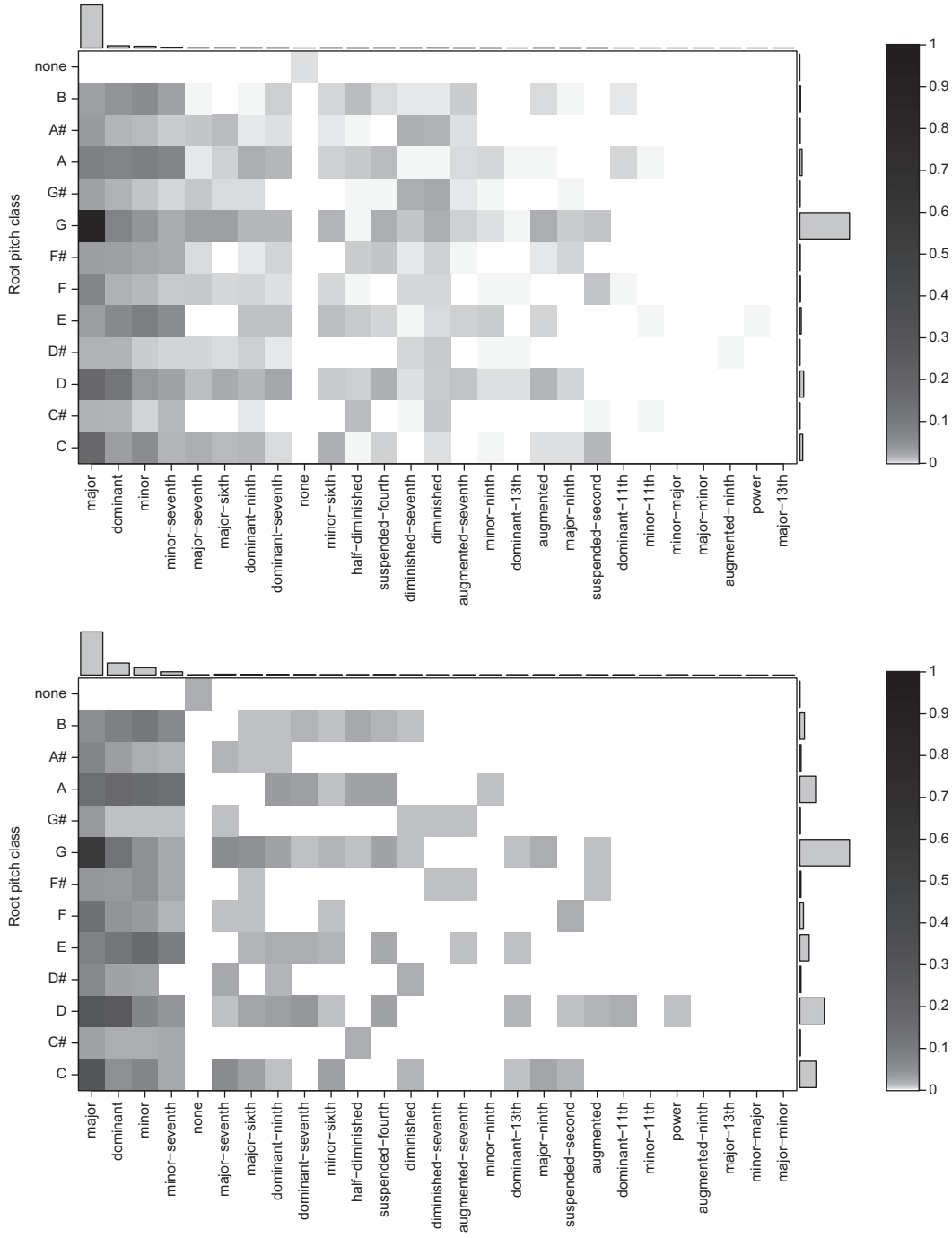

Fig. 5. Conditional distribution of chords $\mathrm{P}_1(C_t|C_{t-1})$ (chord transition probability) for 1-beat frames (top) and 16-beat frames (bottom), plotted for the previous chord C_{t-1} = G-major. The plot is accompanied with linear-scale bar plots of chord root and chord type distributions on the sides.

makes the chord distribution significantly sparser. This means that a system with longer frames will have a more informative melody model, because a single frame will contain more melody notes.

3.5 Interpolation coefficients

The values of the generatively trained interpolation coefficients a_i and b_i are presented in Figure 8. The coefficients for linear and log-linear interpolation follow a similar trend: the bigram model is given a progressively lower weight as the frame length increases, while the melody model behaves in an opposite manner. Indeed, for large frames the bigram model becomes less informative because there is less chordal movement: the longer the frame, the bigger the dominance of the tonic in it (a chord spanning the whole song will have to be the tonic!). At the same time the melody model becomes more informative due to the larger melodic context.

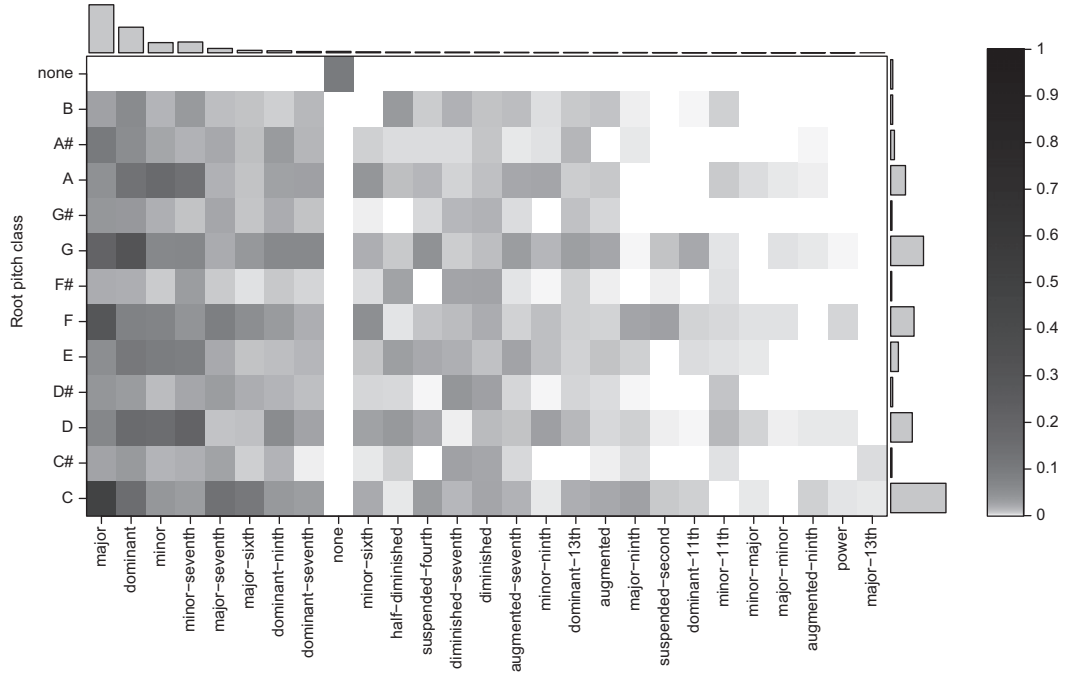

Fig. 6. Conditional distribution of chords $\mathrm{P}_1(C_t|T_t)$ for 1-beat frames, plotted for the tonality of T_t = C-maj. The plot is accompanied with linear-scale bar plots of chord root and chord type distributions on the sides.

4. Evaluation

Due to a large range of goals in the existing literature, we can observe a variety of ways of evaluating harmonization algorithms: through theoretic evaluation of the modelling power by means of cross-entropies (Allan & Williams, 2005; Paiement et al., 2006), through comparison of generated chord sequences with ground-truth chord annotations (Chuan & Chew, 2007; Chuan, 2011), through comparison of single predicted chords with ground-truth (Cunha & Ramalho, 1999), or through subjective listening tests (Lee & Jang, 2004; Simon et al., 2008). Because of the novelty of the proposed solutions, as well as that of the field of automatic harmonization itself, many papers did not offer any evaluation (Ebcioğlu, 1986; Gang et al., 1998; Phon-Amnuaisuk & Wiggins, 1999; Pachet & Roy, 2001). In this paper, we have chosen to perform two complementary evaluations: we first use the theoretic cross-entropy-based evaluation as in Allan and Williams (2005), and Paiement et al. (2006) as a convenient way to validate our interpolation-based approach and we then perform an objective evaluation of the generated chord sequences in the same manner as in Chuan (2011). The code of our algorithm is available online at http://versamus.inria.fr/software-and-data/harmonization.tbz2.

Out of the 2000 Wikifonia lead sheets, 100 lead sheets were used as a test set, 100 as a validation set, and the rest were used as a training set.

4.1 Cross-entropy

An efficient way of determining the modelling power of a model is to compute the normalized negative log-likelihood, or *cross-entropy* (Jurafsky & Martin, 2008). For base-2 logarithms, it can be interpreted as the average number of bits (b) required to encode a single chord symbol (Shannon's optimal code length). So, naturally, the smaller the value, the higher the prediction power. The cross-entropy is calculated from the test chord, melody and tonality sequences **C**, **M** and **T** as

$$\begin{aligned}\mathrm{H}(\mathbf{C}) &= -\frac{1}{T}\log_2 \mathrm{P}(\mathbf{C}|\mathbf{M}, \mathbf{T}, \Lambda) \\ &= -\frac{1}{T}\sum_{t=1}^{T}\log_2 \mathrm{P}(C_t|\mathbf{C}_{1:t-1}, \mathbf{X}_{1:t}),\end{aligned} \tag{14}$$

where Λ denotes the model parameters and T is the number of frames in the test set. The cross-entropy is upper bounded by the cross-entropy of the non-informative uniform chord prior, which in our case is equal to

$$\mathrm{H}_U = -\log_2 \frac{1}{N} = 8.46\ \mathrm{b}, \tag{15}$$

where $N = 351$ is the number of distinct chord labels. The cross-entropies obtained for different frame lengths (from 1/3 to 16 beats) and different generatively trained log-linear combinations of the bigram (B), tonality (T) and melody (M) models are plotted in the upper part of Figure 9. We can observe that the predicting power of the melody (M) and tonality (T) models, although better than the prior alone, is poor for small frame lengths (about 5 bits/frame), but improves by about 0.5 bits/frame as the frame length increases. This is logical, as both the tonality and the chord are musical quantities that depend on a much wider context than a single melody note. However, using large frame lengths limits the

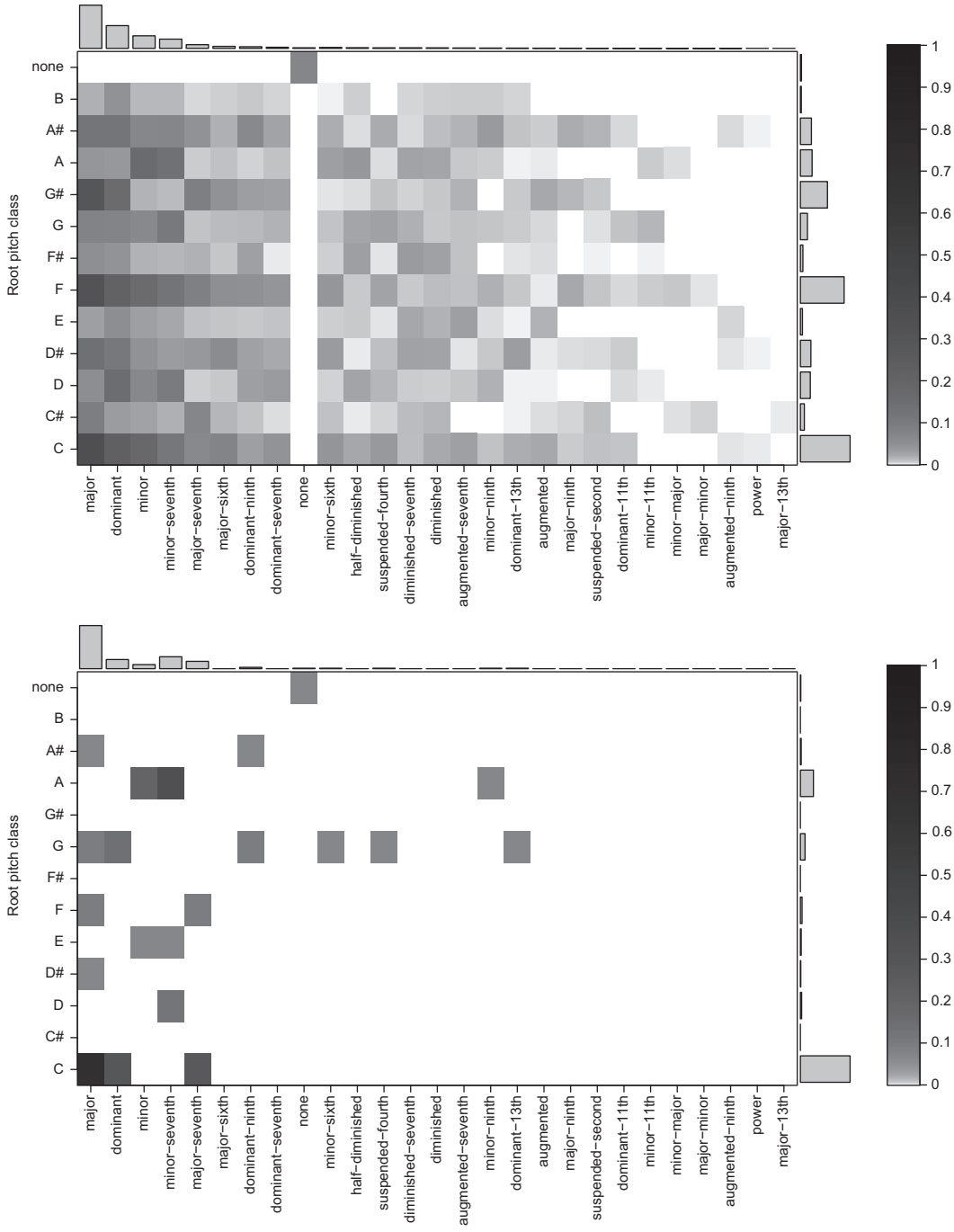

Fig. 7. Conditional distribution of chords $P_3(C_t|M_t)$ for 1-beat frames, plotted for a single C note (top) and for a time frame containing three notes: C, E and G (bottom). Each plot is accompanied with linear-scale bar plots of chord root and chord type distributions on the sides.

temporal precision of the chord estimation, since chords can typically change on any beat (though typically on the down beat). We also observe the benefit of model interpolation: the combined melody and tonality (M + T) model is better than either of the two models alone, and the combined melody and tonality and bigram (M + T + B) model is better than each of these three models alone and than the M + T combination. For 2-beat frames, the latter improvement is equal to 0.37 bit/frame (11%).

Still in the upper part of Figure 9, we can see that the cross-entropy decreases monotonically with decreasing frame length for those models that include the bigram chord progression model, namely B and M + T + B. In fact, it would decrease asymptotically to zero for infinitesimal frame lengths, because predicting the next chord given the current one would be getting easier: one would simply have to predict the same chord and that prediction would be increasingly correct. Therefore we found it useful to complement this plot with a plot of

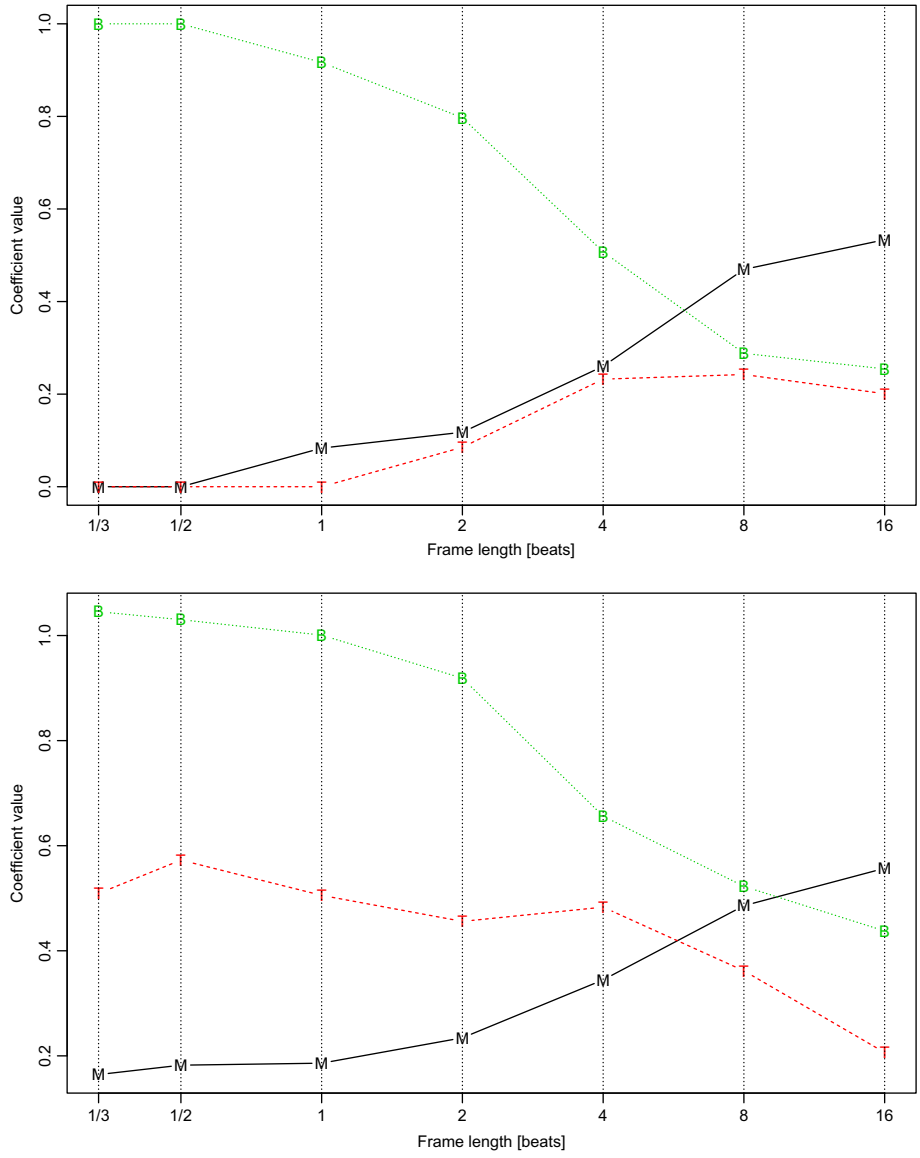

Fig. 8. Values of the generatively trained interpolation coefficients for linear (top) and log-linear (bottom) interpolation. 'B' stands for the bigram model, 'M' for the melody model and 'T' for the tonality model.

cross-entropies normalized per beat instead of per frame in the bottom part of Figure 9. Per-beat cross-entropies are proportional to the total amount of information in the entire dataset, given the model (because the number of beats in the dataset is fixed, as opposed to the number of frames). From that plot we can see that we actually get more informative bigram models for larger frame lengths. On the other hand, this evaluation measure is biased towards larger frame lengths since it is bounded by the per-beat cross-entropy of the uniform chord distribution $H_U/B = (\log_2 N)/B$, where B is the number of beats per frame, which decreases asymptotically towards zero as B increases.

In the light of the above discussion we conclude that the best frame length for harmonization is in the range of few beats, where the cross-entropy is between 2 bits/frame for 1-beat frames and 4 bits/frame for 8-beat frames.

Figure 10 shows the reduction of cross-entropy achieved by log-linear interpolation with respect to linear interpolation ($H_{lin} - H_{loglin}$). Although log-linear interpolation is more time consuming (due to the need for re-normalization by Z), it offers significantly higher modelling power when several models are combined: up to 0.32 bit/frame (10%) for the M + T model and up to 0.26 bit/frame (6.5%) for the M + T + B model.

4.2 Harmonization

In a second experiment we compare the chord sequence generated by the full discriminatively trained model M + T + B with the ground truth chord labels in the test lead sheet files. In this experiment, the timing of the ground truth chord sequence is preserved and does not depend on the chosen frame length.

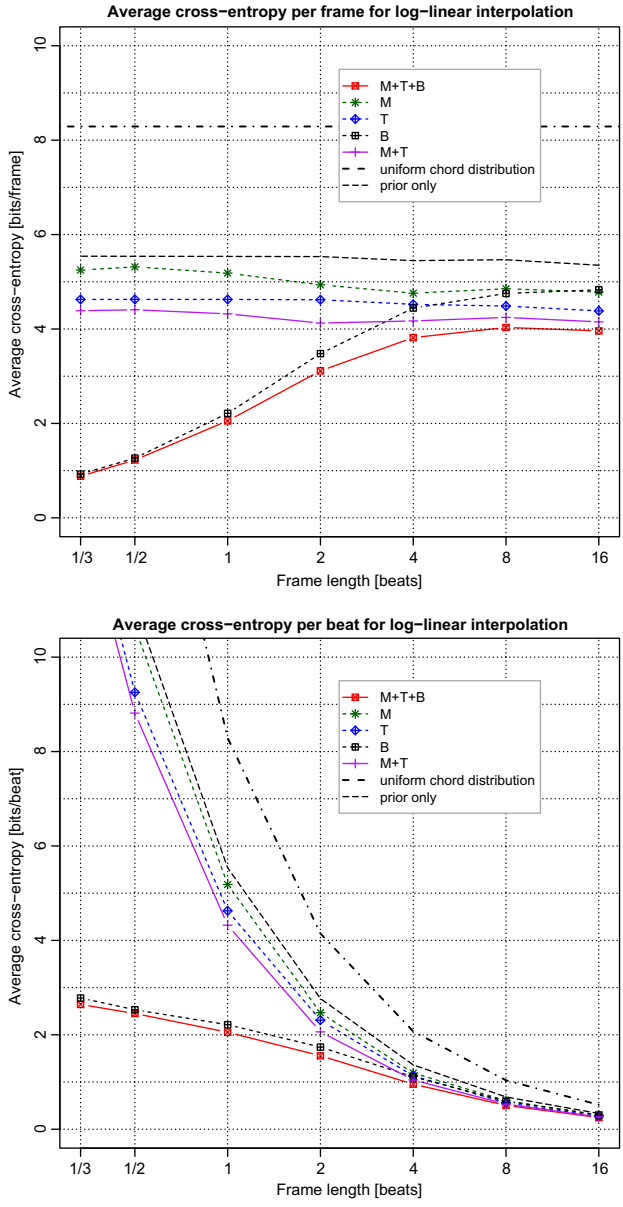

Fig. 9. Cross-entropies calculated for the test dataset, normalized per time frame (top) and per beat (bottom). M stands for the melody model, T for the tonality model and B for the bigram chord progression model. Models were combined using log-linear interpolation.

Chords were estimated via a Viterbi-like algorithm, which finds the most likely sequence of chords given the melody and tonality. The tonality was assumed to be known, because often several tonal interpretations are possible (Aiello & Sloboda, 1994) and we want the resulting chord sequences to be comparable to the ground truth. Similarly to the MIREX competition (Downie, Ehmann, Bay, & Jones, 2010; MIREX community, 2012), the estimated chord types were subsequently clustered into a smaller number of triads (major, minor, augmented, diminished, suspended second or suspended fourth). We compare the chords either in terms of their root note only or in terms of their root note and their triad chord type using two alternative root accuracy measures: a binary one and

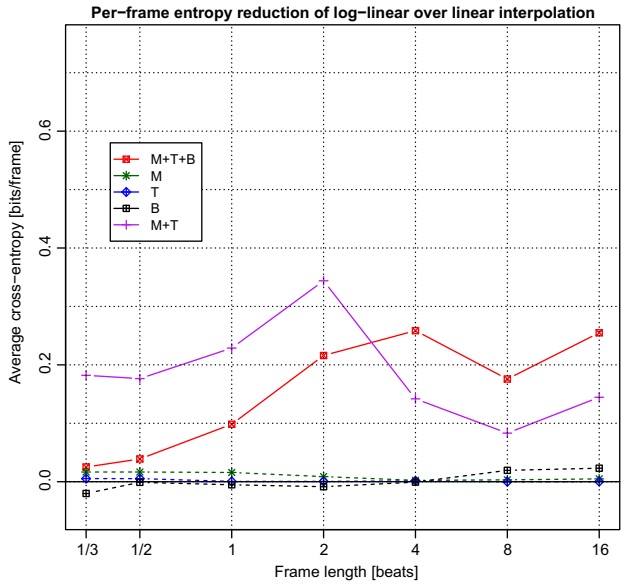

Fig. 10. Cross-entropy reduction of log-linear over linear interpolation.

a weighted one. The latter uses the following, heuristically determined weights: 1 for correct root pitch class estimation, 0.5 for 5 or 7 semitone errors (perfect fourth or fifth), 0.3 for 3, 4, 8 or 9 semitone errors (minor or major third and minor or major sixth), 0.1 for 2 or 10 semitone errors (major second or minor seventh) and 0 for other errors.

For comparison, we have used the results of the state-of-the-art rule-based lead sheet-like harmonization system of Temperley and Sleator, which is freely available on-line at Temperley and Sleator, (2012). The lead sheets were converted to the input format of that algorithm with a time precision ('BaseUnit') of a dotted sixty-fourth note and the metrical structure was generated based on the time signatures and the upbeat (anacrusis) durations extracted from the MusicXML files. It must be noted that the two algorithms had somewhat different design goals (the system of Temperley and Sleator was designed for harmonic analysis in general, not harmonization of melodies) and used different kinds of input and output data, which makes the results difficult to compare. Apart from using the above-mentioned metrical structure, Temperley and Sleator's algorithm estimates only the chord roots, not types, and works with relative, not absolute, chord labels. It also does not use the information about the song's key, as we do in our experiments.

The results for different frame lengths are plotted in Figures 11–13. For all evaluation metrics, the log-linearly interpolated models offer better accuracy than the linear ones and the best results are most often achieved for a frame length of two beats. For shorter frame lengths the melody provides less information, while for longer frame lengths the temporal resolution of the generated chord sequence becomes too coarse. For that frame length, the proposed algorithm with log-linear interpolation outperforms the reference algorithm by 5.5% absolute (17% relative) in terms of root note accuracy and by 4.1% absolute in terms of weighted root note accuracy.

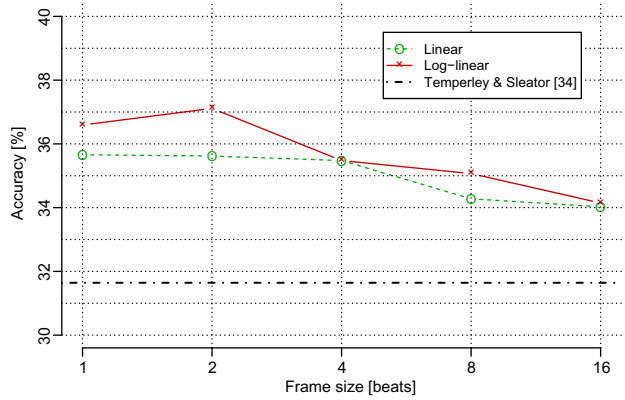

Fig. 11. Root note accuracy obtained for different model frame lengths.

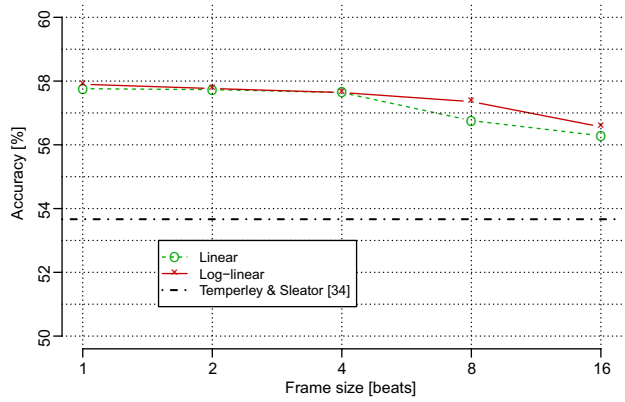

Fig. 12. Weighted root note accuracy obtained for different model frame lengths.

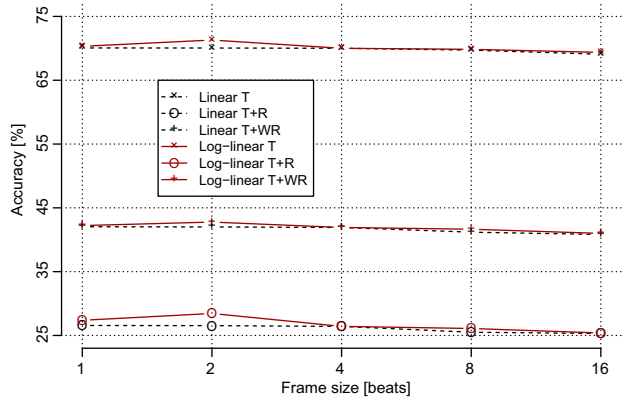

Fig. 13. Triad accuracies obtained for different model frame lengths: triad chord type alone (T), chord type and root note (T + R), and chord type and weighted root note (T + WR).

For 2-beat analysis frames, the log-linearly interpolated models performed better on 63 files, while the reference method was better on the remaining 37 melodies. This was mostly because our method does not take into account the rhythmical structure of the songs, like it does in the reference method, which results in exceedingly long-lasting chords, but also abrupt chord transitions. This can be avoided in future work by adding rhythmical models.

5. Conclusion

In this paper we have presented a novel method of building versatile statistical models of chords for harmonization by joining multiple simpler sub-models by means of linear or log-linear interpolation. To test this idea, we have trained and combined in this way three sub-models: the tonality, the melody and the chord bigram model. We have evaluated the resulting interpolated models in terms of their cross-entropy and observed that log-linear interpolation yields a model whose cross-entropy is lower than the best of the component models and also better than that achieved by linear interpolation. We have then performed a series of harmonization experiments, where we have observed that the proposed log-linearly interpolated model offers higher root chord accuracy than the reference rule-based harmonizer from Temperley and Sleator (1999) by up to 5% absolute.

In future work, a larger number of more complex sub-models, such as relative chord model or a beat model, could be investigated for further improvement in terms of chord accuracy. The proposed method should also be tested on a larger population of songs that would include more diverse musical genres. Subjective listening tests could also be used to analyse the quality of the harmonizations in more detail. The proposed model could also be used for harmonic analysis with musical data consisting of full songs, not only the melody. Finally, the model interpolation methodology could be applied to other music information retrieval tasks that would potentially benefit from modelling several musical aspects simultaneously.

Acknowledgement

This work was performed while the first and third author were with INRIA, F-35042 Rennes Cedex, France, and was supported by INRIA under the Associate Team Program VERSAMUS (http://versamus.inria.fr/).

References

Aiello, R., & Sloboda, J. (1994). *Musical Perceptions*. New York and Oxford: Oxford University Press.

Allan, M., & Williams, C. (2005). Harmonising chorales by probabilistic inference. *Advances in Neural Information Processing Systems*, *17*, 25–32.

Chen, S., Beeferman, D., & Rosenfeld, R. (1998). Evaluation metrics for language models. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, Virginia.

Chuan, C. (2011). A comparison of statistical and rule-based models for style-specific harmonization. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, Miami, FL, pp. 221–226.

Chuan, C., & Chew, E. (2007). A hybrid system for automatic generation of style-specific accompaniment. In *Proceedings of the 4th International Joint Workshop on Computational Creativity*, pp. 57–64. London, UK.

Cunha, U., & Ramalho, G. (1999). An intelligent hybrid model for chord prediction. *Organised Sound*, *4*, 115–119.

Downie, J., Ehmann, A., Bay, M., & Jones, M. (2010). The music information retrieval evaluation exchange: Some observations and insights. In *Advances in Music Information Retrieval* (pp. 93–115). Berlin: Springer.

Ebcioğlu, K. (1986). An expert system for chorale harmonization. In *Proceedings of the National Conference in Artificial Intelligence (AAAI), Philadelphia*, PA (pp. 784–788). Palo Alto, CA: AAAI Press.

Ebcioğlu, K. (1988). An expert system for harmonizing four-part chorales. *Computer Music Journal*, *12*, 43–51.

Fukayama, S., Nakatsuma, K., Sako, S., Nishimoto, T., & Sagayama, S. (2010). Automatic song composition from the lyrics exploiting prosody of the Japanese language. In *Proceedings of the 7th Sound and Music Computing Conference (SMC)*, Barcelona, Spain, pp. 299–302.

Gang, D., Lehman, D., & Wagner, N. (1998). Tuning a neural network for harmonizing melodies in real-time. In *Proceedings of the International Computer Music Conference (ICMC)*, Ann Arbor, USA.

Jaakkola, T., Diekhans, M., & Haussler, D. (2000). A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, *7*, 95–114.

Jebara, T. (2004). *Machine Learning: Discriminative and Generative* (Vol. 755). Berlin: Springer.

Jelinek, F., & Mercer, R. (1980). Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, Amsterdam, The Netherlands pp. 381–397.

Jurafsky, D., & Martin, J. (2008). *Speech and Language Processing*. Engelwood Cliffs, NJ: Prentice Hall.

Klakow, D. (1998). Log-linear interpolation of language models. In *Proceedings of the 5th International Conference on Spoken Language Processing*, Sydney, Australia, pp. 1695–1698.

Lee, H., & Jang, J. (2004). i-Ring: A system for humming transcription and chord generation. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)* Taipei, Vol. 2, pp. 1031–1034.

Meredith, D. (2007). *Computing pitch names in tonal music: a comparative analysis of pitch spelling algorithms* (PhD thesis), University of Oxford, UK.

MIREX community. (2012). *Music Information Retrieval Evaluation eXchange*. http://www.music-ir.org/mirex/wiki/MIREX_HOME (retrieved August)

Och, F., & Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proceeedings of the 40th Annual Meeting on Association for Computational Linguistics*, Philadelphia, PA, pp. 295–302.

Pachet, F., & Roy, P. (2001). Musical harmonization with constraints: A survey. *Constraints*, *6*, 7–19.

Paiement, J., Eck, D., & Bengio, S. (2006). Probabilistic melodic harmonization. In *Proceedings of the 19th Canadian Conference on Artificial Intelligence*, London, Ontario, pp. 218–229.

Papadopoulos, H., & Peeters, G. (2007). Large-scale study of chord estimation algorithms based on chroma representation and HMM. In *Proceedings of the IEEE International Workshop on Content-Based Multimedia Indexing (CBMI)*, Bordeaux, France, pp. 53–60.

PG Music Inc. (2012). *Band-in-a-Box*. http://www.pgmusic.com/ (August).

Phon-Amnuaisuk, S., & Wiggins, G. (1999). The four-part harmonisation problem: a comparison between genetic algorithms and a rule-based system. In *Proceedings of the Artificial Intelligence and Simulation of Behavior Conference* (Vol. 99, pp. 28–34). London: AISB.

R Development Core Team. (2011). R: A Language and Environment for Statistical Computing [Computer software manual]. Vienna, Austria. Retrieved from http://www.R-project.org/

Raphael, C., & Stoddard, J. (2004). Functional harmonic analysis using probabilistic models. *Computer Music Journal*, *28*, 45–52.

Rathinavelu, C., & Deng, L. (1996). The trended HMM with discriminative training for phonetic classification. In *Proceedings of the 4th International Conference on Spoken Language (ICSLP)* (Vol. 2, pp. 1049–1052). Piscataway, NJ: IEEE Press.

Rennie, J., & Rifkin, R. (2001). *Improving multiclass text classification with the support vector machine* (Technical Report No. AIM-2001-026). Cambridge, MA: Massachusetts Institute of Technology.

Rothgeb, J. (1979). Simulating musical skills by digital computer. In *Proceedings of the Annual ACM Conference* (pp. 121–125). New York: ACM Press.

Rothgeb, J.E. (1969). *Harmonizing the unfigured bass: A computational study*. (PhD thesis), Yale University, USA.

Scholz, R., Vincent, E., & Bimbot, F. (2009). Robust modeling of musical chord sequences using probabilistic *n*-grams. In *Proceedinggs of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, pp. 53–56.

Simon, I., Morris, D., & Basu, S. (2008). MySong: automatic accompaniment generation for vocal melodies. In *Proceedings of the 26th SIGCHI Conference on Human Factors in Computing Systems*, Florence, Italy, pp. 725–734.

Steels, L. (1986). Learning the craft of musical composition. In *Proceedings of the International Computer Music Conference (ICMC)*, Den Haag, The Netherlands, pp. A-27-A-31.

Temperley, D., & Sleator, D. (1999). Modeling meter and harmony: A preference-rule approach. *Computer Music Journal*, *23*, 10–27.

Temperley, D., & Sleator, D. (2012). *Harmonic Analyzer*.http://www.cs.cmu.edu/~sleator/harmonic-analysis/ (August)

Wikifonia Foundation. (2012). *Wikifonia*. http://www.wikifonia.org/ (August)

Winograd, T. (1968). Linguistics and the computer analysis of tonal harmony. *Journal of Music Theory*, *12*, 2–49.

Woodland, P., & Povey, D. (2002). Large scale discriminative training of hidden Markov models for speech recognition. *Computer Speech & Language*, *16*, 25–47.

Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, *22*, 179–214.