Chapter 8
Interactive Melodic Analysis

David Rizo, Plácido R. Illescas, and José M. Iñesta

Abstract In a harmonic analysis task, melodic analysis determines the importance and role of each note in a particular harmonic context. Thus, a note is classified as a harmonic tone when it belongs to the underlying chord, and as a non-harmonic tone otherwise, with a number of categories in this latter case. Automatic systems for fully solving this task without errors are still far from being available, so it must be assumed that, in a practical scenario in which the melodic analysis is the system's final output, the human expert must make corrections to the output in order to achieve the final result. Interactive systems allow for turning the user into a source of high-quality and high-confidence ground-truth data, so online machine learning and interactive pattern recognition provide tools that have proven to be very convenient in this context. Experimental evidence will be presented showing that this seems to be a suitable way to approach melodic analysis.

David Rizo
Universidad de Alicante, Alicante, Spain
Instituto Superior de Enseñanzas Artísticas de la Comunidad Valenciana (ISEA.CV), EASD Alicante, Alicante, Spain
e-mail: drizo@dlsi.ua.es

Plácido R. Illescas
Universidad de Alicante, Alicante, Spain
e-mail: placidoroman@gmail.com

José M. Iñesta
Universidad de Alicante, Alicante, Spain
e-mail: inesta@dlsi.ua.es

8.1 Introduction

Musical analysis is the means to go into depth and truly understand a musical work. A correct musical analysis is a proper tool to enable a musician to perform a rigorous and reliable interpretation of a musical composition.

It is also very important for music teaching. In addition, the outcome of computer music analysis algorithms is very relevant as a first step for a number of music information retrieval (MIR) applications, including similarity computation (de Haas, 2012; Raphael and Stoddard, 2004), reduction of songs to an intermediate representation (Raphael and Stoddard, 2004), music summarization (Rizo, 2010), genre classification (Pérez-Sancho et al., 2009), automatic accompaniment (Chuan and Chew, 2007; Simon et al., 2008), pitch spelling in symbolic formats (Meredith, 2007), algorithmic composition (Ulrich, 1977), harmonization (Ebcioğlu, 1986; Feng et al., 2011; Kaliakatsos-Papakostas, 2014; Pachet and Roy, 2000; Raczyński et al., 2013; Suzuki and Kitahara, 2014), performance rendering (Ramírez et al., 2010), preparing data for Schenkerian analysis (Kirlin, 2009; Marsden, 2010), key finding (Temperley, 2004), metre analysis (Temperley and Sleator, 1999), and others.

From the artificial intelligence perspective, the interest in studying how a machine is able to perform an intrinsically human activity is a motivation by itself (Raphael and Stoddard, 2004). Furthermore, from a psychological point of view, the comparison of analyses by a computer with those made by a human expert may yield interesting insights into the process of listening to musical works (Temperley and Sleator, 1999).

The first written evidence of a musical analysis dates from 1563 and appears in a manuscript entitled 'Praecepta Musicae Poeticae' by Dressler (Forgács, 2007). In 1722, Jean-Philippe Rameau, in his 'Traité de l'harmonie réduite à ses principes naturels', established the basis of harmonic analysis (Rameau, 1722). However, music analysis enjoyed a significant growth in the 19th century.

From the computational point of view, the various aspects of musical analysis have all been addressed since the 1960s (Forte, 1967; Rothgeb, 1968; Winograd, 1968), and there has been a sustained interest in the area up to the present day. In the last few years, several theses (bachelor, master and Ph.D.) have been published from this point of view (de Haas, 2012; Granroth-Wilding, 2013; Mearns, 2013; Sapp, 2011; Tracy, 2013; Willingham, 2013), which underlines the importance of this area of study.

The relevance of a melodic analysis depends on its ultimate purpose: in composition it helps the author to study the different harmonization options, or in the reverse direction, given a chord sequence, to create melodic lines. In the case of analysing a work for playing or conducting, it helps to establish the role each note plays regarding stability or instability. For teaching, it is an indispensable tool for the student and the teacher.

The analysis of a composition involves several interrelated aspects: aesthetic analysis related to the environment of the composer that influences him or her when creating the work, formal analysis to suitably identify the structure of the composition and its constituent elements, and finally tonal analysis, which can be divided into harmonic and melodic analysis. Harmonic analysis studies chords and tonal functions, to shed light on the tensions and relaxations throughout a work, while melodic analysis establishes the importance and role of each note and its particular harmonic context.

This chapter is focused on melodic analysis, specifically using a symbolic format as input. Thus, as output, every note in a musical work is classified as a *harmonic tone* when it belongs to the underlying chord, and as a *non-harmonic tone* otherwise, in which case it should be further assigned to a category, such as *passing tone*, *neighbour tone*, *suspension*, *anticipation*, *echappée*, *appoggiatura* and so on (see Willingham (2013, p. 34) for a full description).

There is still no objective benchmark or standardized way of comparing results between methods. Even if such a benchmark existed, very different analyses can be correctly obtained from most musical works, a fact that reflects different analysts' preferences (Hoffman and Birmingham, 2000).

Nonetheless, it is widely accepted that none of the computerized systems proposed to date is able to make an analysis that totally satisfies the musicologist or musician; and what is worse, it seems that no system can be built to totally solve the problem. The case of melodic analysis is a good example of the variability between the different interpretations that can be extracted from a piece of music, due to the fact that it depends on harmony, which in turn is derived from parts (such as accompaniment voices) that may not be available or that may not even exist when making the analysis.

Maxwell (1984) differentiated between "computer-implemented" analysis, where the output of the system is the final analysis, and "computer-assisted" analysis, where the output must be interpreted by the user. All systems found in the literature[1] choose the "computer-implemented" analysis approach. In order to overcome the limitation exposed above, we introduce a system that follows the "computer-assisted" approach—that is, an interactive melodic analysis, integrating automatic methods and interactions from the user. This is accomplished in the present work by using the "Interactive Pattern Recognition" (IPR) framework, which has proven successful in other similar tasks from the human action point of view, like the transcription of hand-written text images, speech signals, machine translation or image retrieval (see Toselli et al. (2011) for a review of IPR techniques and application domains). We will present experimental evidence that shows that IPR seems to be a suitable way to approach melodic analysis.

This chapter is structured as follows. First the main trends in harmonic analysis, along with ways of dealing with melodic analysis, and the introduction of interactivity, are reviewed in Sect. 8.2. The classical pattern matching classification paradigm, most commonly used so far, is formulated in Sect. 8.3. The interactive pattern recognition approach will then be introduced in Sect. 8.4.

Our proposal to solve the problem of melodic analysis using various approaches based on manual, classical pattern matching and IPR methods will be described in Sect. 8.5. A graphical user interface (GUI) has been developed to assert the expectations presented theoretically, and it is described in Sect. 8.6. The experimental

[1] Except the study by Taube and Burnson (2008), but that work focuses on the correction of analyses rather than on assisting the analyst's task.

results are then presented in Sect. 8.7, and finally, some conclusions are drawn in Sect. 8.8.

8.2 State of the Art

Several non-comprehensive reviews of computational harmonic analysis can be found in the recent literature (de Haas, 2012; Kröger et al., 2010; Mearns, 2013).

Two main tasks in harmonic analysis are recurrent in most of the approaches: first the partition of the piece into segments with harmonic significance, then the assignment of each segment to a chord in a key context using either a Roman numeral academic approach (e.g., V7 dominant chord) or a modern notation (e.g., a chord like GMaj7). From a human perspective, an analysis cannot be made as a sequence of independent tasks (e.g., first a key analysis, then a chordal analysis, then a melodic analysis and so on). However, the simultaneity in the execution of these phases may depend on the particular musical work. In some cases all the tasks are computed simultaneously, while in others, for each phase, several possibilities are generated and the best solution has to be selected using an optimization technique. For example, melodic analysis conditions the other tasks, helping in discarding ornamental notes that do not belong to the harmonic structure, in order to make decisions on segmentation and chord identification.

8.2.1 Segmentation

The partition of a piece of music into segments with different harmonic properties (i.e., key, chord, tonal function), referred to as "one of the most daunting problems of harmonic detection" by Sapp (2007, p. 102), has been tackled so far using two related approaches: one that may be named *blind*, because it does not use any prior tonal information, and another that takes into account some computed tonal information from the beginning, that Mouton and Pachet (1995) have called *island growing*. The *blind* approach is based only on timing information and involves chopping the input into short slices (Barthélemy and Bonardi, 2001; Illescas et al., 2007; Pardo and Birmingham, 2000), using either points of note onset and offset, a given fixed duration, or the duration of the shortest note in a bar or in the whole piece. Then, once the key and chord information are available after the initial segmentation, these slices are combined, if they are contiguous and share the same chord and key, to build meaningful segments (usually in a left-to-right manner).

The *island-growing* method finds tonal centres based on evident chords, cadences or any clue that allows a chord to be attached to a given segment in a key context. Once these tonal centres are obtained, they are grown in a similar way to the *blind* approach. This is a more usual approach in the literature (Meredith, 1993; Sapp, 2007; Scholz et al., 2005; Ulrich, 1977). Note that this method also needs to split the work horizontally in order to assign these tonal centres, so distinguishing between *blind* and *island growing* in some cases is difficult or not totally clear.

Finally, as Pardo and Birmingham (2002) state, there are approaches that receive an already segmented input (e.g., Winograd, 1968) or where it is not clear how the segmentation is obtained.

8.2.2 Knowledge-Based and Statistical Approaches

The identification of chords and keys alone, given the already computed segments or simultaneously with the calculation of these segments, has been performed using two very different approaches: one based on rules established by experts, sometimes referred to as *knowledge-based*, and the other built on top of statistical machine learning systems, which Chuan and Chew (2011) properly refer to as *data-driven*.

There is no sound experimental evidence on which approach yields the best analysis results, but currently it seems to be assumed that machine learning systems are more adequate than knowledge-based systems (Chuan and Chew, 2011). Some systems use a hybrid solution. Nevertheless, even the less knowledge-based systems incorporate at least some a priori information in the intermediate music representation itself or in the learning strategy designed from a preconceived guided solution. Some of them even include some rules that restrict or direct the statistical methods (Raphael and Stoddard, 2004).

Of the two approaches, knowledge-based systems were the first to be used to tackle the problem. They were formulated using preference-rule systems (Temperley, 1997, 2001; Temperley and Sleator, 1999), using a classical forward-chaining approach or other typical solutions in expert systems (Maxwell, 1984; Pachet, 1991; Scholz et al., 2005), as constraint-satisfaction problems (Hoffman and Birmingham, 2000), embedded in the form of grammars (de Haas, 2012; Rohrmeier, 2007; Tojo et al., 2006; Winograd, 1968) or using numerical methods based on template matching. The latter methods work by matching the input set of pitches that comes from the segmentation process to a list of possible chord templates. By using a similarity measure between chords, the list of templates is ordered, and the algorithm either selects the most similar template or passes the list to a later process that uses either some kind of algorithm (Prather, 1996; Taube, 1999) or an optimization technique to find the best sequence of chords by means of a graph (Barthélemy and Bonardi, 2001; Choi, 2011; Illescas et al., 2007; Kirlin, 2009; Pardo and Birmingham, 2002). Passos et al. (2009) use a *k*-nearest neighbours technique to perform the matching process.

The main advantage of statistical machine learning systems is their ability to learn from examples, either supervised from tagged corpora or unsupervised, thus, theoretically overcoming the problem of the variability of the myriad of applicable rules. There are in the literature almost as many proposals for this approach as there are machine learning techniques: *HMPerceptron* to solve a supervised sequential learning (SSL) problem, like those used in part-of-speech tagging (Radicioni and

Esposito, 2007), hidden Markov models (Mearns, 2013; Passos et al., 2009; Raphael and Stoddard, 2004) or neural networks (Scarborough et al., 1989; Tsui, 2002).

Both approaches have advantages and disadvantages, as noted in various studies (Mouton and Pachet, 1995). The main disadvantage of rule-based systems is the impossibility for any system to include rules for every possible situation, able to cope, for example, with any genre or composer. In fact, in many situations, composers try to break established rules in a creative manner. Another disadvantage of rule-based approaches is the fact that, in many cases, two different rules may conflict. This situation has often been solved by using preference rules (*meta-rules*) that solve those conflicts. Raphael and Stoddard (2004) highlight another problem, namely, that, as the rule systems work by ordering a sequence of decisions, the propagation of errors from an early decision may compromise the final result. The main advantage of rule-based systems is their capacity for explanation, which may be used to guide the user action in an interactive approach or educational environment. In the case of numerically based methods, Raphael and Stoddard (2004) point out that the numerical values returned by their chord similarity algorithm are difficult to justify and must be found just by empirically tuning the system. To overcome this problem, statistical procedures have been applied that automatically optimize parameter values by methods like linear dynamic programming (Raphael and Nichols, 2008) or genetic algorithms (Illescas et al., 2011).

Besides segmentation and chord identification, there are important details that differentiate the depth of the different studies reported in the literature. One is the handling of modulations and tonicizations. Modulation is the process by which one tonal centre is substituted by another. Usually, the tonality may change throughout a piece. In many cases, it starts with a key, modulates to other keys and eventually returns to the initial tonality. The concept of tonicization (Piston, 1987) is used to describe the cadence of a secondary dominant onto its tonic, in such a way that, in a given tonality, when there is a perfect cadence onto any degree, this degree acts as the tonic of the secondary dominant that precedes it. More detailed explanations are provided by Tsui (2002, pp. 7–8) and Mearns (2013, pp. 20–22). Some methods consider tonicization to be just a key change, ignoring this temporal key context change (Illescas et al., 2007), others reinterpret the result in a post-process to adapt it to the correct interpretation (Kirlin, 2009). There are, however, plenty of approaches that explicitly include this concept in their models (Hoffman and Birmingham, 2000; Rohrmeier, 2011; Sapp, 2011; Scholz et al., 2005; Taube, 1999).

8.2.3 Melodic Analysis

The other aspect that is central to the present work is melodic analysis. No work has focused in depth just on melodic tagging in a harmonic analysis task from a computational point of view. A first attempt was made by Illescas et al. (2011) and a musicological study was presented by Willingham (2013). Nevertheless, in many studies, melodic analysis has received the attention it deserves (e.g., Chuan and

Chew, 2011; Mearns, 2013; Sapp, 2007) or, at least, it has been acknowledged that a better understanding of melodic analysis would improve the chord identification process (Pardo and Birmingham, 2002; Raphael and Stoddard, 2004). In some methods, ornamental notes are removed in an a priori manual preprocess, in order to avoid the melodic analysis task (Winograd, 1968). In many studies, notes are chosen just using their metrical position: that is, strong notes, or using a regular span (Yi and Goldsmith, 2007). Others use very simple rules: for example, Barthélemy and Bonardi (2001) and Kirlin (2009) assume that non-chord notes are followed by a joint movement. In rule-based systems, there are usually rules that deal specifically with melodic analysis, e.g., Temperley's (2001) "Ornamental Dissonance Rule" or rules 10 to 20 in Maxwell's (1984) model. Template matching was used by Taube (1999).

From a machine learning perspective, two contemporary approaches have been proposed that work in virtually the same way: one proposed by the authors of the current work (Illescas et al., 2011) that will be extended here, and Chuan and Chew's (2011) "Chord-Tone Determination" module. In both cases, notes are passed as a vector of features (up to 73 in Chuan and Chew's (2011) model; whereas Illescas et al. (2011) use a smaller but similar set) to a decision tree learner that learns rules to classify either harmonic tones vs. non-harmonic tones (Chuan and Chew, 2011) or harmonic tones vs. each different kind of non-harmonic tone (Illescas et al., 2011).

8.2.4 Interactivity

One of the aspects of this work that has received less attention in the literature is the opportunity for interaction between potential users and such a system. Some authors have expressed in some cases the need for interactivity (Scholz et al., 2005) that is implicit in the concept of "computer-assisted" analysis suggested by Maxwell (1984). Sapp (2011) reviews errors generated by his algorithm, finding that sometimes the obtained key was wrong but closely related to the actual tonic key. From a classical standpoint, this is an error, but maybe it could be considered a minor mistake. In an interactive approach, this could easily be solved by presenting a ranking of keys to the user. Phon-Amnuaisuk et al. (2006) present their system as a "platform for music knowledge representation including harmonization rules to enable the user to control the system's harmonization behaviour". This "user control" is indeed an interactive process. Something similar is asserted by Taube (1999): "The user may directly control many aspects of the analytical process".

Some authors have expressed their intention to add an interactive user interface; for example, Chuan and Chew (2010) present a preliminary design. For a harmonization task, Simon et al. (2008) add some possible interaction that allows the user to choose the kind of chords generated. In the teaching environment, the system "Choral Composer" (Taube and Burnson, 2008) allows the students to see their mistakes as they do each exercise (guided completion).

Other software tools for visualizing musical analyses include Chew and François' (2003) "MuSA.RT, Opus 1", which represents a work using the Spiral Array model; and the graphical user interface tool, "T2G", cited by Choi (2011).[2]

There is also the "Impro-Visor" software,[3] which is a music notation program designed to help jazz musicians compose and hear solos similar to ones that might be improvised. The system, built on top of grammars learned from transcriptions, shows improvisation advice in the form of visual hints.

Finally, though not interactive, the "Rameau" system (Passos et al., 2009) allows users to experiment with musicological ideas in a graphical visualization interface, and Sapp's (2011) "keyscapes" also provide visual analyses of works.

The interactive pattern recognition paradigm has not been applied to the tonal analysis task so far. However, many of the problems uncovered when analysing the analyses performed by computer tools (see for example the manual analysis of errors by Pardo and Birmingham (2002)) could be addressed in an interactive model. Any data-driven approach can directly benefit from the IPR approach as well. It would not be straightforward, but adding user decisions as specific rules to a model, in a similar manner to that used in a case-based-reasoning system (Sabater et al., 1998), could be a way to take advantage of user feedback.

The lack of standardized ground truth or evaluation techniques has been mentioned above. Some methods compare their results using very isolated works. Nevertheless, it seems that J. S. Bach's harmonized chorales have been frequently used as a corpus (Illescas et al., 2007, 2008, 2011; Maxwell, 1984; Radicioni and Esposito, 2007; Tsui, 2002), perhaps because they form the most scholastic corpus available and because most analysts agree upon how these pieces should be analysed.

Regarding evaluation techniques, there is no agreement on a quantitative evaluation measure to use in order to compare the performance of different proposals. In any case, as will be detailed below, under the interactive pattern recognition approach introduced here, systems are not assumed to be fully automatic but rather to require user supervision. Here, quantitative evaluation is therefore less oriented to performance accuracy and more to the workload (e.g., number of user interactions) that is required in order to achieve the correct output.

8.3 Classical Pattern Recognition Approach

The computational methods utilized in the present work for solving the problem of melodic analysis are related to the application of pattern recognition and matching techniques to the classification of the notes in a score into seven categories: harmonic and six classes of non-harmonic tone. This way, we can consider this task as a classical seven-class classification problem in pattern recognition. For that, we can consider that every note is an input sample, x_i. From the sample and its context

[2] http://members.shaw.ca/akochoi-T2/jazz-harmonic-analysis/index.html

[3] http://www.cs.hmc.edu/ keller/jazz/improvisor/

(x_{i-1}, x_i, x_{i+1}), a number of features can be computed that are expressed as a feature vector, $\mathbf{x}_i$, that can be regarded as evidence for categorizing the note i. From this information, the system's underlying model $\mathcal{M}$ should be able to output a hypothesis $\hat{h}_i$, classifying the input sample into one of the seven classes.

Usually, $\mathcal{M}$ is inferred from example pairs $(\mathbf{x}, h) \in \mathcal{X}$ provided to the system in the training phase. For learning, a strategy for minimizing the error due to incorrect h is followed. Once the system is trained by achieving an acceptable error measure, the model is applied to new, previously unseen, samples. In this operation phase, the decision on each sample is the hypothesis $\hat{h}_i$ that maximizes the posterior probability value estimated $\Pr(h_i \mid \mathbf{x}_i)$, considering that this value is provided by the model learnt:

$$\hat{h}_i = \arg\max_{h \in \mathcal{H}} \Pr(h \mid \mathbf{x}_i) \approx \arg\max_{h \in \mathcal{H}} P_{\mathcal{M}}(h \mid \mathbf{x}_i) \ . \tag{8.1}$$

The input to the classification system is a series of vectors $\mathbf{x} = \mathbf{x}_1, ..., \mathbf{x}_{|M|}$, where $|M|$ is the number of notes of the melody. The output is a sequence of decisions $\mathbf{h} = h_1, ..., h_{|M|} \in \mathcal{H}$ = {H,P,N,S,AP,AN,ES} (see Sect. 8.5 for a definition of these classes).

8.4 Interactive Pattern Recognition Approach

Multimodal human interaction has become an increasingly important field that aims at solving challenging application problems in multiple domains. Computer music systems have all the potential features for this kind of technique to be applied: multimodal nature of the information (Lidy et al., 2007), need for cognitive models (Temperley, 2001), time dependency (Iñesta and Pérez-Sancho, 2013), adaptation from human interaction (Pérez-García et al., 2011) and so on.

Assuming that state-of-the-art systems are still far from being perfect, not only in terms of accuracy, but also with respect to their applicability to any kind of music data, it seems necessary to assume that human intervention is required, at least for a correction stage after the automatic system output. It could also be interesting to take advantage of this expert knowledge during the correction process and to work on techniques for efficiently exploiting the information provided (that relies on the user's expertise) in the context of adaptive systems. Therefore, the pattern recognition (PR) system accuracy is just a starting point, but not the main issue to assess. In IPR systems, evaluation tries to measure how efficiently the system is taking advantage of this human feedback and to work on techniques towards better adaptive schemes able to reduce the user's workload.

Placing the human in the IPR framework requires changes in the way we look at problems in these areas. Classical PR is intrinsically grounded on error-minimization algorithms, so they need to be revised and adapted to the new, minimum-human-effort performance criterion (Toselli et al., 2011). This new paradigm entails important research opportunities involving issues related to managing the feedback information provided by the user in each interaction step to improve raw performance, and the use of feedback-derived data to adaptively re-train the system and tune it to the user behaviour and the specific data at hand.

We shall now analyse these aspects of research in IPR in more detail in the context of our research.

8.4.1 Exploiting Feedback

We have described the solution to our problem as a hypothesis $\hat{\mathbf{h}}$ coding the classes of every note in our problem score. These hypotheses were those that maximize the posterior probabilities among all possible hypotheses for every note. Now, in the interactive scheme, the user observes the input $\mathbf{x}$ and the hypothesis $\hat{\mathbf{h}}$ and provides a feedback signal, f, in the form of a local hypothesis that constrains the hypothesis domain $\mathcal{H}$, so we can straightforwardly say that $f \in \mathcal{H}$. Therefore, by including this new information in the system, the best system hypothesis now corresponds to the one that maximizes the posterior probability, but given the data and the feedback:

$$\hat{\mathbf{h}} = \arg\max_{h \in \mathcal{H}} P_{\mathcal{M}}(\mathbf{h} \mid \mathbf{x}, f) \;, \tag{8.2}$$

and this can be done with or without varying the model $\mathcal{M}$. After the new hypothesis is computed, the system may prompt the user to provide further feedback information in a new interaction step, k. This process continues until the system output, $\hat{\mathbf{h}}$, is acceptable to the user.

Constructing the new probability distribution and solving the corresponding maximization, may be more difficult than the corresponding problems with feedback-free posterior distributions. The idea is to perform the analysis again after each feedback input, f_k, taking this information as a constraint on the new hypothesis in such a way that the new $\hat{\mathbf{h}}^{(k+1)} \in \mathcal{H}^{(k+1)} = \mathcal{H}^{(k)} - \hat{\mathbf{h}} \subset \mathcal{H}^{(k)}$.[4] This way, the space of possible solutions is restricted by the user's corrections, because the user is telling the system that the hypothesis $\hat{\mathbf{h}}$ is not valid. Clearly, the more feedback-derived constraints can be added, the greater the opportunity to obtain better hypotheses.

This iterative procedure can make available a history of hypotheses, $h' = \hat{\mathbf{h}}^{(0)}, \hat{\mathbf{h}}^{(1)}$, ..., $\hat{\mathbf{h}}^{(k)}$, from previous interaction steps that lead eventually to a solution that is acceptable to the user. Taking this into account explicitly as

$$\hat{\mathbf{h}}^{k+1} = \arg\max_{h \in \mathcal{H}} P_{\mathcal{M}}(\mathbf{h} \mid \mathbf{x}, h', f) \;, \tag{8.3}$$

may improve the prediction accuracy gradually throughout the correction process.

[4] In order to simplify the notation we have omitted that vector $\hat{\mathbf{h}}$ is actually a member of the Cartesian product $\mathcal{H}^{|M|}$.

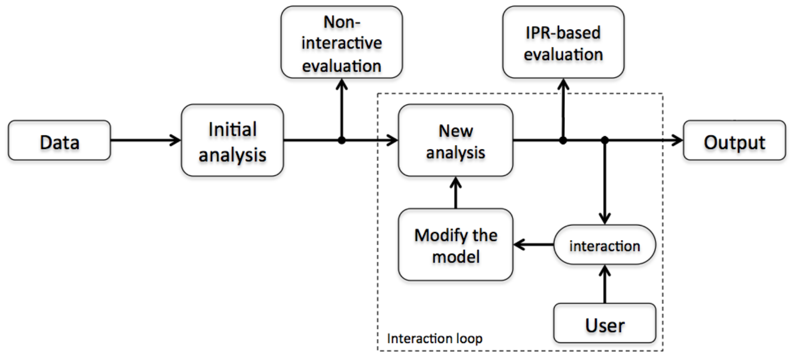


Fig. 8.1 Performance and evaluation based on an interactive pattern recognition (IPR) approach

8.4.2 System's Adaptation from Feedback

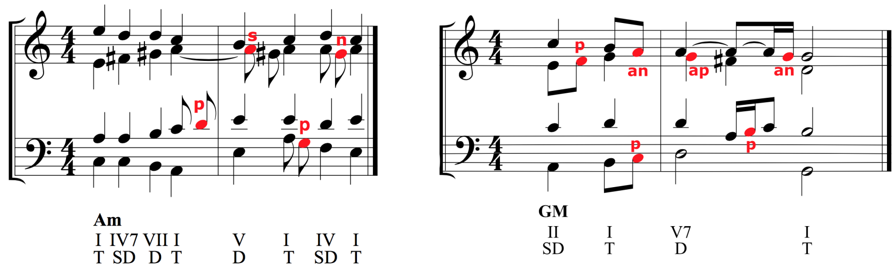
Human interaction offers a unique opportunity to improve a system's behaviour by tuning the underlying model. Everything discussed in the preceding section can be applied without varying the model $\mathcal{M}$, restricting the solution space through the feedback and thus approximating the solution.

We can go one step further using the feedback data obtained in each step of the interaction process f_k, which can be converted into new, valid training information, $(\mathbf{x}_i, h = f_k)$. This way, after each correction we get a new training set $\mathcal{X}^{(k+1)} = \mathcal{X}^{(k)} \cup \{(\mathbf{x}_i, h = f_k)\}$, allowing for the model to be re-trained or adapted. After a number of iterations the initial training set $\mathcal{X}^{(0)}$ has been completed with ground-truth training pairs.

The application of these ideas in our musical analysis framework will require establishing adequate evaluation criteria. These criteria should allow the assessment of how adaptive training algorithms are taking the maximum advantage of the interaction-derived data to ultimately minimize the overall human effort.

The evaluation issue in this interactive framework is different from classical PR algorithms (see Fig 8.1). In those systems, performance is typically assessed in terms of elementary hypothesis errors; i.e., by counting how many local hypotheses h_i differ from the vector of correct labels (non-interactive evaluation in Fig. 8.1). For that, the assessment is based on labelled training and test corpora that can be easily, objectively, and automatically tested and compared, without requiring human intervention in the assessment procedures.

Nevertheless, in an interactive framework, a human expert is embedded "in the loop", and system performance has to be gauged mainly in terms of how much human effort is required to achieve the goals. Although the evaluation of the system performance in this new scenario apparently requires human work and judgement, by carefully specifying goals and ground truth, the corpus-based assessment paradigm is still applicable in the music analysis task, just by counting how many interaction



(a) BWV 286, 2nd bar (b) BWV 422, bars 12 and 13

Fig. 8.2 Examples of non-harmonic notes in a melodic analysis. Only non-harmonic notes are tagged

steps are needed to produce a fully correct hypothesis (see IPR-based evaluation in Fig. 8.1).

8.5 Method

The problem we address here is the melodic analysis of a work in a tonal context—in particular, to tag all notes as harmonic tone (H), passing tone (P), neighbour tone (N), suspension (S), appoggiatura (AP), anticipation (AN), or echappée (ES) (see Fig. 8.2). As described in Sect. 8.2, this process, embedded in a more general tonal analysis problem, has been tackled so far using knowledge-based systems and machine learning techniques. In previous work, using the classical pattern recognition paradigm (Illescas et al., 2011), similar success rates for both approaches were obtained using some of Bach's harmonized chorales, with better results using statistical methods. The IPR paradigm will be applied to improve that result.

The model in IPR systems can be built using any of the classifiers employed in classical PR approaches. In order to assess the improvement of IPR over PR, the same classifier will be used in the experiments for both paradigms.

Machine learning systems are those that can benefit the most from the IPR improvements highlighted above. In order to choose among the variety of machine learning algorithms, only those capable of providing a full explanation of the decisions taken are considered here, with the aim of offering the user a full and understandable interactive experience. This is why a decision-tree learner has been chosen. Illescas et al. (2011) used a RIPPER algorithm (Cohen, 1995) to overcome the imbalance in the data (around 89% of the notes are harmonic tones). However, in agreement with the results of Chuan and Chew (2011), a C4.5 decision tree algorithm (Quinlan, 2014) gave better results using a leave-one-out scheme on a training corpus of 10 Bach chorales (previously used by (Illescas et al., 2011)). We extend and provide details of this corpus in Sect. 8.7.2.1.

8.5.1 Features

The classifier receives as input a note x_i represented by a vector of features, $\mathbf{x}_i$, and yields as output a probability for each tag: $P(h_i \mid \mathbf{x}_i)$, $h_i \in \mathcal{H} = \{\text{H, P, N, S, AP, AN, ES}\}$ on which the classification decision will be made. We shall now define these features.

Definition $\mathit{previousIntervalName}(x_i) \in \mathbb{N}$

The absolute interval of a note with its predecessor as defined in music theory, i.e., unison, minor 2nd, major 2nd, 3rd, etc.

Definition $\mathit{previousIntervalDir}(x_i) = \begin{cases} \mathit{undefined}, & i = 1 \\ \mathit{ascending}, & \mathit{pitch}(x_i) > \mathit{pitch}(x_{i-1}) \\ \mathit{descending}, & \mathit{pitch}(x_i) < \mathit{pitch}(x_{i-1}) \\ \mathit{equal}, & \mathit{pitch}(x_i) = \mathit{pitch}(x_{i-1}) \end{cases}$

Definition $\mathit{previousIntervalMode}(x_i) \in \{\mathit{major}, \mathit{minor}, \mathit{perfect}, \mathit{augmented}, \mathit{diminished}, \mathit{double\ augmented}, \mathit{double\ diminished}\}$

This is computed using the music theory rules from the $\mathit{previousIntervalName}$ and the absolute semitones from x_{i-1} to x_i.

Definition *nextIntervalName, nextIntervalMode and nextIntervalDir* are defined similarly using the interval of the note x_{i+1} with respect to x_i.

Definition $\mathit{tied}(x_i) \in \mathbb{B}$ *is true if the note* x_i *is tied from the note* x_{i-1}.

Definition $\mathit{rd}(x_i) = \mathit{duration}(x_i)/\mathit{duration}(\mathit{beat})$

The relative duration function determines the ratio between the duration of x_i and the duration of a beat.

Definition $\mathit{ratio}(x_i) = \frac{\mathit{rd}(x_i)}{\mathit{rd}(x_{i-1})} \times \frac{\mathit{rd}(x_i)}{\mathit{rd}(x_{i+1})}$

The ratio function is used to compare the relative duration of x_i to its next and previous notes.

Definition $\mathit{meterNumerator}(x_i)$ is the numerator of the active metre at $\mathit{onset}(x_i)$. The value of $\mathit{onset}(\cdot)$ is defined locally for each measure, depending on the metre, as the position in the measure in terms of sixteenth notes, counted from 0 to (16 $\times$ numerator / denominator) – 1.

Definition $\mathit{instability}(x_i)$: given $\mathit{onset}(x_i)$, and $\mathit{meterNumerator}(x_i)$, it returns a value relative to the metrical weakness of x_i.

The stronger the beat in which the onset of a note is, the lower its instability value will be. See Table 8.1 for the list of values used.[5]

[5] The instability values for the binary metres can be obtained directly using the method described by Martin (1972). Ternary and compound metres need a straightforward extension of the method.

Table 8.1 Instability values as a function of the onset position for the different metres used. The resolution is one sixteenth note

Metre	Instability values indexed by $onset(x_i)$
4/4	(1, 9, 5, 13, 3, 11, 7, 15, 2, 10, 6, 14, 4, 12, 8, 16)
2/4	(1, 5, 3, 7, 2, 6, 4, 8)
3/4	(1, 7, 4, 10, 2, 8, 5, 11, 3, 9, 6, 12)
6/8	(1, 5, 9, 3, 7, 11, 2, 6, 10, 4, 8, 12)
9/8	(1, 7, 13, 4, 10, 16, 2, 8, 14, 5, 11, 17, 3, 9, 15, 6, 12, 18)
12/8	(1, 9, 17, 5, 13, 21, 3, 11, 19, 7, 15, 23, 2, 10, 18, 6, 14, 22, 4, 12, 20, 8, 16, 24)

Definition $\mathit{nextInstability}(x_i) = \mathit{instability}(x_{i+1})$; refers to the instability of the next note.

Definition $\mathit{belongsToChord}(x_i) \in \mathbb{B}$ is true if, given the pitch class of the note $pc(x_i)$, at $onset(x_i)$ there is an active chord made up of a set of notes $\mathbf{C}$, and $pc(x_i) \in \mathbf{C}$.

Definition $\mathit{belongsToKey}(x_i) \in \mathbb{B}$ is true if, given the pitch class $pc(x_i)$, at $onset(x_i)$ there is a key using the scale made up of a series of notes $\mathbf{S}$, and $pc(x_i) \in \mathbf{S}$.

The scale is the major diatonic for major keys, and the union of ascending, descending, and harmonic scales for minor keys.

Definition $\mathit{prevNoteMelodicTag}(x_i) \in \mathcal{H}$ is the melodic tag of the previous note, h_{i-1}, if already analysed.

Definition $\mathit{nextNoteMelodicTag}(x_i)$ is equivalent to the previous definition but referred to the next note, h_{i+1}.

The information about key and chord needed in the definitions above depends on the order in which the user carries out the different analysis stages. If, at a given stage, any of this information is not available, a feature will remain undefined, and the classifier will not yet be able to use it. During the interaction stage, this information becomes increasingly available.

Note that this feature-extraction scheme is using a window size of 3 notes. In some studies (e.g., Meredith, 2007) a wider window is used for determining the pitch spelling of notes. However, in our case, our system is able to explain the decision using the predecessor and successor notes, based on the underlying harmony, as explained in most music theory books.

8.5.2 Constraint Rules

As we are just focusing on the baroque period, some rules have been manually added that constrain the set of possible outputs by removing those that are invalid (e.g., two consecutive anticipations). Moreover, these rules allow the system to take advantage of some additional information the user provides by using the system, as will be seen below.

As introduced above, the system avoids invalid outputs by checking the following conditions. Let x_i be the note to be analysed, $pc(x_i)$ its pitch class, c the active chord at $onset(x_i)$, and $\mathbf{C}$ the pitches in chord c:

1. x_i cannot be tagged as H (harmonic tone) if its onset occurs on a weak beat, i.e., $instability(x_i) > meterNumerator(x_i)$, and $pc(x_i) \notin \mathbf{C}$.
2. h_i = H always if $pc(x_i) \in \mathbf{C}$.
3. x_i cannot be tagged as passing tone (P) if $h_{i-1} \in \{\mathrm{AP}, \mathrm{S}, \mathrm{AN}, \mathrm{N}\}$ (appoggiatura, suspension, anticipation or neighbour tone).
4. x_i cannot be tagged as N if $h_{i-1} \in \{\mathrm{AP}, \mathrm{S}, \mathrm{AN}, \mathrm{P}\}$.
5. x_i cannot be tagged as $\{\mathrm{A}, \mathrm{AP}, \mathrm{S}\}$ if $h_{i-1} \in \{\mathrm{AP}, \mathrm{S}, \mathrm{AN}, \mathrm{N}, \mathrm{P}\}$.

These rules involving key and chord information, as well as the tagging of surrounding notes, are only available to the system through the interactive action of the user. The computing of key and chord information would imply the full tonal analysis process, and this work focuses only the melodic analysis task, the rest of the process is performed manually by the user.

8.5.3 IPR Feedback and Propagation

The underlying classification model was required to provide a readable explanation of the decision mechanism, so we focus on decision trees, as discussed at the beginning of Sect. 8.5. The C4.5 decision tree algorithm, using the same features both for the classical PR approach and the IPR system, was utilized. The C4.5 algorithm provides the a posteriori probability $P(h_i \mid \mathbf{x}_i)$ as the proportion of samples in the leaf that belongs to each class (Margineantu and Dietterich, 2003) using a Laplacian correction to smooth the probability estimations. Although it cannot be incrementally updated, it trains in a very short time. In this way, in our case, it is fully re-trained after each interaction using the new information provided by the user. This fact does not limit its usability for melodic analysis, since the re-training is perceived as a real-time update by the user. Moreover, the size of the data set will never be too large, because analysis rules are specific to each genre, so the need for scalability is not an issue.

As introduced in Sect. 8.4.1, each time the user provides a feedback $f \in \mathcal{H}$, the model is rebuilt as if the pair $(\mathbf{x}_i, h'_i = f)$ was in the training set. Furthermore, this means that, if the user amends the analysis of a note x_i with features $\mathbf{x}_i$ to be $h'_i \neq \hat{h}_i$, the analysis $\hat{h}_j$ of further notes x_j with features $\mathbf{x}_j = \mathbf{x}_i$ should be the same, i.e., the analysis of them will be modified accordingly as $h'_j = h'_i$. This is called *propagation* and it is performed for the rest of notes $x_j, \forall j \neq i$ after each user interaction on note x_i.

8.6 Application Prototype

In order to prove the validity of the IPR approach for the melodic analysis task in a real user scenario and in order to study how it leverages users' effort using the assistant system, an interactive prototype has been developed in JavaFX 8,[6] a graphical user interface developer framework built on top of the Java language.

The application allows not only the melodic analysis, but also helps in the task of key and chord analysis, because chord identification and melodic analysis cannot be done as isolated tasks, but need to be done in a coordinated fashion. The reason is that the decision as to which notes have to be included to form a chord depends on which ones have been tagged as harmonic; but in order to tag a note as harmonic, one has to predict which chord will be formed (as well as other considerations).

In order to perform the analysis, the prototype has the following features:

- It reads and writes from and to MusicXML. Chords are encoded using the corresponding schema elements, the remaining analyses, such as tonal functions, tonicizations, and so on, are encoded/decoded using lyrics.
- It reads from `**kern` format including the harmonic spines.
- It renders the score visually allowing for the selection of individual elements.
- It helps the user select the most probable chord and key at each moment.
- It permits the introduction and editing by the user of all the tonal analysis: melodic tags, chords, key changes, tonicizations, and secondary dominants.
- It logs all the user actions for later study.

In order to compare the user's actions using the three approaches considered (manual, automatic PR-based automatic, and IPR-assisted), the user can select the operation mode in the application.

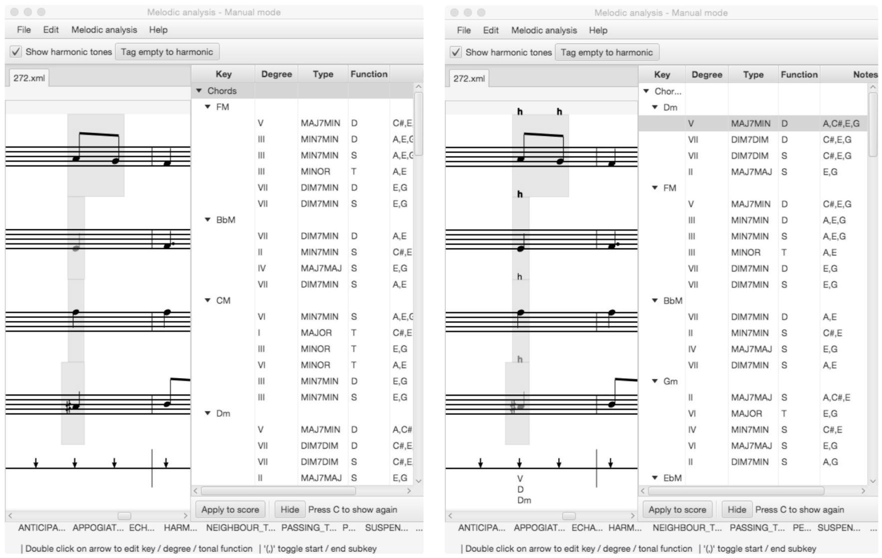
8.6.1 Manual Mode

Not too different from employing a sheet of paper and a pencil, one computer-aided way of doing an analysis is to use any kind of score editor like Finale or Musescore, adding the melodic tags as lyrics under each note. This approach, which was adopted by the authors in their creation of the first ground truth (Illescas et al., 2007), is tedious, and the effort required to analyse a work, measured as number of interactions, is at least equal to the number of notes. That method is not taken into account in this experimentation.

The use of the prototype in manual mode allows the user to manually introduce the melodic tag of each note. It acts as a helping tool to annotate the key and chord of the selected sonority in an assisted way. The only logged user actions will be those related to the melodic tagging, with those referring to chord and key being discarded.

In a typical scenario, the user proceeds as follows:

[6] http://www.oracle.com/technetwork/es/java/javafx/overview/index.html



(a) After selecting alto note E, sonority is highlighted. Last key was FM

(b) Apply selected chord. All notes belonging to chord are tagged as ‘H’

Fig. 8.3 Highlight of sonority and application of selected key and chord

1. A note is selected. The corresponding sonority is highlighted accordingly by including all the notes that are simultaneously active at any time during the selected note (Fig. 8.3(a)). For them, a list of possible keys and chords in each key is displayed hierarchically. The details of how this list is constructed are given below.
2. A chord is selected from the list of valid keys and chords and is applied to the current sonority (Fig. 8.3(b)). If the user prefers to apply another chord and key not present in the proposed list (such as tonicizations or secondary dominants, not included in it), it can be done using a dialogue as shown in Fig. 8.4. Once the context is established, as a help to the user, notes not belonging to the active chord are highlighted.
3. Finally, using a set of predefined keyboard keys, the user selects the suitable melodic tag for each note. The system just logs this last action, because it is the only one that corresponds strictly to the melodic analysis task.

This process is repeated for each note in the musical work. Note that the user may backtrack on a decision and the same note could be tagged several times.

In most musical works, at least in the baroque period, almost all notes are harmonic tones, not ornamental. This implies that the note tags follow a highly imbalanced distribution in favour of class H. In order to avoid the user having to carry out unnecessary actions, the prototype includes a button that tags all previously untagged notes as harmonic (see Fig. 8.5). This allows the user to tag only non-harmonic tones, reducing considerably the number of interactions.

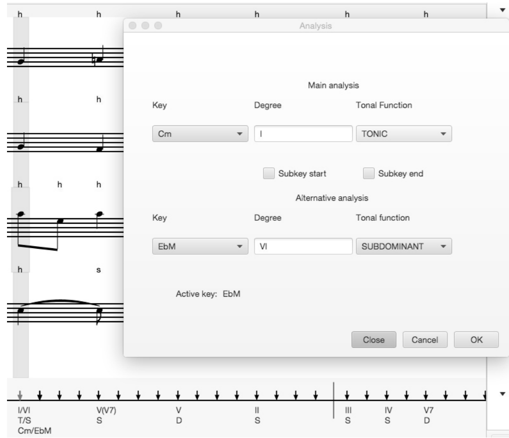


Fig. 8.4 Dialogue box that allows the user to apply a chord not present in the proposed chords list. Used for assigning tonicizations and secondary dominants

8.6.1.1 Chord and Key List Construction

The valid keys added to the list are those whose associated scale includes all the notes in the selected sonority.

The chords are chosen using a template-based approach: given the set of notes, all possible combinations of groups of at least two notes are matched with the list of chord types shown in Table 8.2. Finally, the list of keys is ranked using the following ordering: the current key first (or the major mode of the key present in the key signature if no previous key was found), then the next key up and down in the circle of fifths and the relative minor or major. The rest of the keys are ordered inversely, proportional to the distance along the circle of fifths. In the minor key, the relative major key is located at the second position of the list.

Inside each key, the chords with more notes in the sonority are ordered first. When having the same number of notes, those containing the root are located in upper positions, and when comparing chords containing the root and having the same number of notes, the tonal functions are ordered this way: tonal, dominant, and subdominant. Figure 8.3(b) shows an example.

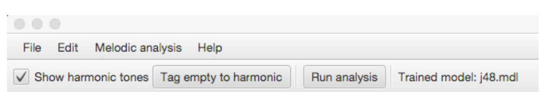


Fig. 8.5 Button that tags all non previously tagged notes as “harmonic tone”

Table 8.2 Chord templates. The semitones of the first pitch correspond to the semitones from the tonic of the chord

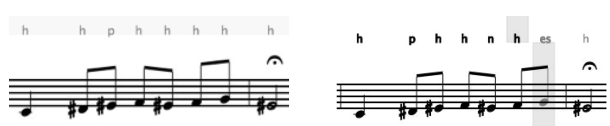
Chord type	Semitones from previous pitch
Major triad	(4,3)
Minor triad	(3,4)
Augmented triad	(4,4)
Diminished triad	(3,3)
Major with minor seventh	(4,3,3)
Augmented with major seventh	(4,4,3)
Diminished with minor seventh	(3,3,4)
Diminished with diminished seventh	(3,3,3)
Major seventh with major seventh	(4,3,4)
Minor seventh with minor seventh	(3,4,3)

8.6.2 Automatic Mode

In automatic mode, previously introduced as "computer-implemented" by Maxwell (1984) and described under the classical pattern recognition paradigm (Sect. 8.3), the user proceeds using this protocol:

1. First, the system analyses the score automatically. The Weka (Hall et al., 2009) implementation of the C4.5 algorithm (Quinlan, 2014) has been embedded in the prototype and it is fed using the features described in Sect. 8.5.1, excluding the chord- and key-related features (*belongsToChord* and *belongsToKey*) because they are not available when the work is analysed automatically the first and only time.
2. All notes have now an analysis tag, that may be correct or not. Now, the user proceeds just like the manual mode explained above, by choosing chords and keys, and, instead of setting the melodic tag for each note, just changing those tags that the C4.5 classifier has misclassified (see Fig. 8.6).

The system has been trained using a bootstrap set of 10 Bach chorales manually tagged (see list of works below in Sect. 8.7.2.1).



(a) Automatic melodic analysis

(b) Melodic analysis correction by user

Fig. 8.6 Highlight of sonority and application of selected key and chord

8.6.3 Assisted Mode

The assisted mode corresponds to the introduced IPR approach, named by Maxwell (1984) as "computer-assisted" analysis. Here the system reacts against all the user actions. The loop of actions is described next:

1. As in the manual mode, the user selects a note and a sonority is highlighted, for which the user identifies and assigns key and chord.
2. The prototype performs a melodic analysis of the work using the C4.5 classifier. Now the features *belongsToChord* and *belongsToKey* already have a value for all the notes located from the selected sonority and forwards. Moreover, all the constraint rules (Sect. 8.5.2) can now be applied.
3. As in the automatic mode, the user may amend (feedback) any melodic analysis tag, which fires the propagation of that decision to all notes with the same features, and runs again the C4.5 classifier, now re-trained with the new corrected sample. A user-amended tag is never modified by the new classifier decision.
4. The process is repeated until all notes are melodically tagged.

This process is not a mere repetition of the automatic mode process for each note, it has several important implications:

- In order to show the valid chords in the help list, notes tagged as any of the non-harmonic tones are not used. This method narrows the search of the desired chord, but also forces the user to tag as harmonic the notes the system had tagged incorrectly as non-harmonic. It may seem that the correct chord and key identification can slow down the melodic tagging. However, as the *belongsToChord* and *belongsToKey* features use the key information, the classifier has more information about the harmonic context after each interaction, which boosts the melodic tagging.
- The change of a melodic tag affects the surrounding notes, that may be modified by the constraining rules after a user interaction, leading to a correction of a possibly incorrect tagging.

This process may not be done sequentially from left to right because the user could proceed in an "island-growing" way, by first locating tonal centres and then browsing back and forth.

8.6.4 User Interaction Analysis

The prototype logs each action carried out by the user. In this study, only the actions relating to the melodic analysis itself have been taken into account. So, in order not to block the user interaction at any moment, the Java logging framework has been customized to export the kind of information shown in Table 8.3, printing the user actions to a file, using a separate thread. This file has been parsed in order to extract

Table 8.3 Example of log entries

Action time stamp	Session time stamp	Action type	Action
		...	
1417009922812	1417009922796	actionloggersystem	started
1417010341734	1417009922796	MELODICANALYSIS.CHANGE	PASSING_TONE
1417010390390	1417009922796	MELODICANALYSIS.CHANGE	PASSING_TONE
1417010550375	1417009922796	MELODICANALYSIS.CHANGE	SUSPENSION
1417010665140	1417009922796	MELODICANALYSIS.CHANGE	HARMONIC
		...	

times and number of user interactions. The system records session information in order that the user may close the prototype and continue the analysis task in a later session.

8.7 Experiments

8.7.1 Proof of Concept: Ability to Learn Interactively

In order to assess the ability of the interactive system to learn online from the interaction of the user, a simulation has been built in which the system receives a musical work whose notes have been manually tagged with their corresponding melodic analysis labels. It simulates the user actions as follows:

1. The work is automatically analysed using a (possibly untrained) classifier. After this step, both the correct tag for each note obtained from the manual analysis and the tag assigned by the classifier are available for every note.
2. The interaction system now proceeds as a human would: it looks for an error in the analysis (at random to simulate the back-and-forth action by a real user), then it replaces the incorrect label (assigned by the classifier) with the correct tag (obtained from the previous manual analysis).
3. This interaction fires the interactive pattern-matching loop, i.e., the feedback decision is propagated to all notes not previously corrected, and the classifier model is updated including this new sample.
4. The process is repeated until no errors remain. The number of changes performed (equal to the number of times the process has been carried out) is the actual system performance evaluation value.

Using this setup, the complete collection of Bach's harmonized chorales (see Sect. 8.7.2.1) has been used. The system has been fed sequentially with the set of chorales, starting with an untrained C4.5 classifier that learns after each user action

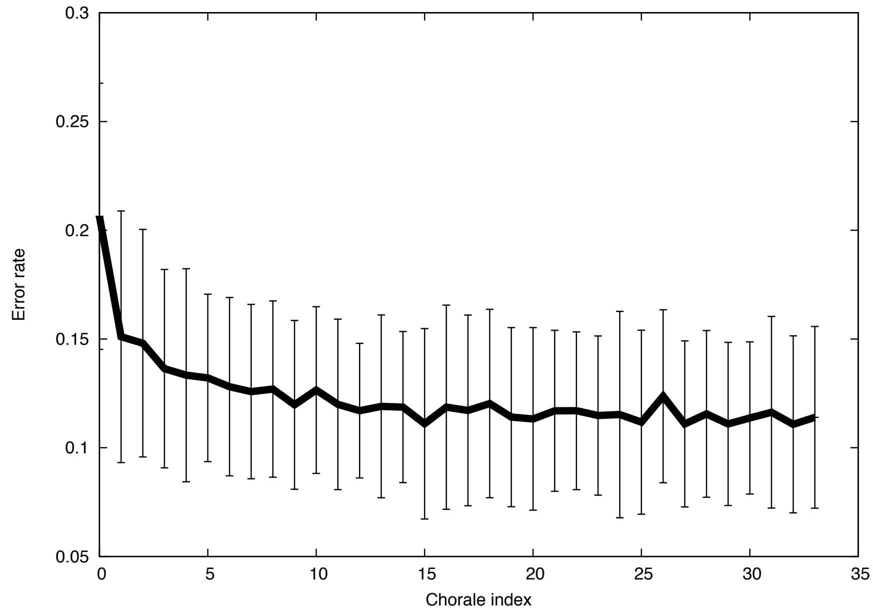


Fig. 8.7 Evolution of the error rate as the user analyses works. The x axis represents the order in which a work is analysed. The plot shows the average (thick line) of the results of 100 different orderings of the input corpus. The standard deviations are also shown. The error rate is measured as the number of interactions required to correct a piece divided by the number of notes it contains

on each musical piece. Not all chorales have the same level of complexity, so the presentation order may affect the evolution of the error rate. In order to avoid any bias, the process has been repeated 100 times in different random orderings of the pieces. Figure 8.7 shows how the error rate becomes lower as the system analyses more and more works. This provides evidence that the system is able to learn from the interaction of the user.

Considering the possibility of using a decision-tree classifier able to update the model online without having to train it for each new feedback sample, the simulation has also been carried out using a Hoeffding tree (VFDT) (Hulten et al., 2001). However, the results obtained, both with the classical PR approach and with the setup just described were worse than those using the C4.5 classifier. Moreover, there was no perceptible improvement in the training speed for the online VFDT learning compared to the full re-training of the C4.5.

8.7.2 Experimental Setup and Data

In order to test the prototype, the following process has been followed:

1. A musicologist[7] has manually tagged the training set.
2. Students of the final course of a music degree analysed the test set using the three modes of the prototype: manual, automatic, and interactive. The analysis of the same work using different modes by the same user was avoided.
3. In all analyses, the system was bootstrapped using the same trained model.

8.7.2.1 Corpora

The system was assessed using some of Bach's chorales, encoded using MusicXML files. These pieces are widely accepted as a scholastic tonal harmony ground truth, as mentioned in Sect. 8.2 above. Furthermore, they contain monodic lines for each voice that help in the construction of features for the classifier.

For the training phase, the following chorales were used: catalogue BWV numbers 89/6, 148/6, 253, 272, 274, 275, 280, 437, 438.

For the test, the following were used: catalogue BWV numbers 255, 256, 257, 259, 260, 281, 282, 285, 286, 287, 288, 290, 292, 293, 294, 295, 296, 420, 421, 423, 424, 426, 427, 429, 431.

BWV 257, 260, 420 were analysed by several students, thus, a total of 30 pieces were utilized for test.

8.7.3 Results

Figure 8.8 shows the results of the melodic analysis using the test set described above. The percentage of less than 15% of non-harmonic tones (NHT) on average, gives an indication of the minimum number of interactions the user has to complete when using the "Tag all as harmonic tones" button first, and then tagging just the non-harmonic tones. This assumes that no note is tagged twice due to a user changing his or her mind.

The results show that the best performance is obtained using the proposed IPR approach. The graph demonstrates what could be expected. The results for the manual approach are worse than the theoretical minimum number of interactions (i.e., $\approx$ 15%), expected to be equal to the number of non-harmonic tones plus one for the action of pushing the "tag all as harmonic" button. This is caused by the fact that the user, solving the problem heuristically, provides several possible solutions leading to different taggings of the same note. The automatic, classical PR approach, leverages the user's effort, who takes advantage of some of the correctly classified melodic tags that, in turn, help with the correct selection of keys and chords, thus narrowing the user's heuristic search for the solution. Finally, in the proposed IPR system the advantages found in the PR approach are used, and they are improved in two ways by the use of the feedback from the user. First, this feedback enriches the

[7] The author Plácido R. Illescas.

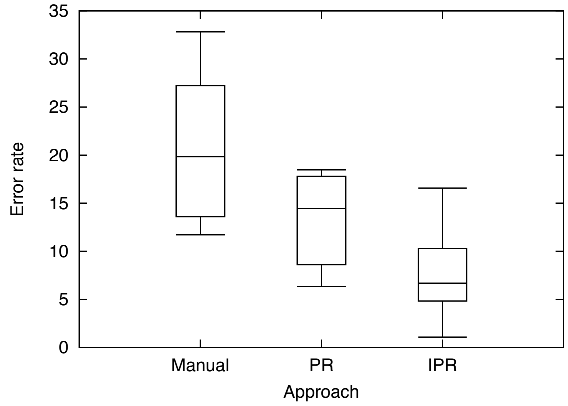


Fig. 8.8 Results in terms of number of user interactions required to solve the melodic analysis. The percentages are obtained as the number of interactions divided by the number of notes

input features to the classifier after each interaction. Second, the feedback re-trains the model, incorporating the corrected analyses as new examples in the training set. Moreover, these examples are very probably found in the same work several times, reducing in this way the number of interactions required to get a correct final analysis.

It is important to stress the fact that the IPR approach in this task has a competitive advantage over a single PR one, due to the fact that the harmonic information (key context and chords) cannot be analysed before doing the melodic analysis. This is because each of the three aspects depends on the other two. Thus, the PR approach cannot use the harmonic information that the user provides with his or her interactive use of the tool.

In the current setup of the experiment, the most important improvement from the user feedback comes from the use of contextual information added after each user interaction, which is not available for the classical PR paradigm. The propagation, which is the same as incorporating new samples in the model, that can be used to solve similar situations in the future, constitutes a second enhancement. Finally, the experiments have not really taken full advantage of the on-line training and tuning of the model, as the model is not incrementally maintained from analysis to analysis. Indeed, it is reset for each experiment in order to be compared with the manual and PR approaches under similar conditions.

8.8 Conclusions

The tonal analysis of musical works in a computational environment is a problem that has been addressed since the 1960s. Many different approaches have been proposed since that time that roughly fall into two categories belonging to the pattern recognition discipline: knowledge-based and machine learning methods. Although interesting results have been obtained in several studies, a full solution to the problem has still not been found.

In this work, a different approach, called Interactive Pattern Recognition (IPR), has been applied, that focuses on trying to reduce the final effort made by the user, rather than minimizing the errors initially made by the system in an automatic analysis.

Using well-known techniques from the classical Pattern Recognition paradigm, IPR improves their performance by incorporating user feedback into the model after each interaction, which helps the system to be refined as the user works with it.

In order to explore the suitability of the method, this IPR approach has been applied to one part of the tonal analysis task: melodic analysis, leaving aside the key and chord computation to be done manually. The proposal has been assessed by means of a prototype software application that, besides performing this melodic analysis using IPR, helps the user to annotate the tonal context of the work.

Using a widely-used corpus, a subset of Bach's harmonized chorales, the IPR paradigm has been proven to provide a suitable approach for finally obtaining a satisfactory solution to the problem of tonal analysis, assisted by computer from the user's point of view.

Acknowledgements This work has been funded by the Spanish Ministry project TIMuL (TIN2013-48152-C2-1-R). Additional support was provided by European FEDER funds. Thanks to Craig Sapp and David Meredith for their valuable aid and advice.

References

Barthélemy, J. and Bonardi, A. (2001). Figured bass and tonality recognition. In *Proceedings of the Second International Symposium on Music Information Retrieval (ISMIR 2001)*, pages 129–136, Bloomington, IN.

Chew, E. and François, A. (2003). MuSA.RT: Music on the spiral array. Real-time. In *Proceedings of the Eleventh ACM International Conference on Multimedia*, pages 448–449, New York, NY.

Choi, A. (2011). Jazz harmonic analysis as optimal tonality segmentation. *Computer Music Journal*, 35(2):49–66.

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

Chuan, C.-H. and Chew, E. (2010). Quantifying the benefits of using an interactive decision support tool for creating musical accompaniment in a particular style. In *Proceedings of the Eleventh International Society for Music Information Retrieval Conference (ISMIR 2010)*, pages 471–476, Utrecht, The Netherlands.

Chuan, C.-H. and Chew, E. (2011). Generating and evaluating musical harmonizations that emulate style. *Computer Music Journal*, 35(4):64–82.

Cohen, W. W. (1995). Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning (ICML)*, pages 115–123.

de Haas, W. B. (2012). *Music information retrieval based on tonal harmony*. PhD thesis, Utrecht University.

Ebcioğlu, K. (1986). An expert system for chorale harmonization. In *Proceedings of the American Association for Artificial Intelligence, AAAI*, pages 784–788.

Feng, Y., Chen, K., and Liu, X.-B. (2011). Harmonizing melody with meta-structure of piano accompaniment figure. *Journal of Computer Science and Technology*, 26(6):1041–1060.

Forgács, R. (2007). *Gallus Dressler's Praecepta Musicae Poeticae*. University of Illinois Press.

Forte, A. (1967). Music and computing: The present situation. In *AFIPS Proceedings of the Fall Joint Computer Conference*, pages 327–329, Anaheim, CA.

Granroth-Wilding, M. (2013). *Harmonic analysis of music using combinatory categorial grammar*. PhD thesis, University of Edinburgh, UK.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. (2009). The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18.

Hoffman, T. and Birmingham, W. (2000). A constraint satisfaction approach to tonal harmonic analysis. Technical report, Electrical Engineering and Computer Science Department, University of Michigan.

Hulten, G., Spencer, L., and Domingos, P. (2001). Mining time-changing data streams. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 97–106. ACM Press.

Illescas, P. R., Rizo, D., and Iñesta, J. M. (2007). Harmonic, melodic, and functional automatic analysis. In *Proceedings of the 2007 International Computer Music Conference (ICMC)*, pages 165–168, Copenhagen, Denmark.

Illescas, P. R., Rizo, D., and Iñesta, J. M. (2008). Learning to analyse tonal music. In *Proceedings of the International Workshop on Machine Learning and Music (MML 2008)*, pages 25–26, Helsinki, Finland.

Illescas, P. R., Rizo, D., Iñesta, J. M., and Ramírez, R. (2011). Learning melodic analysis rules. In *Proceedings of the International Workshop on Music and Machine Learning (MML 2011)*.

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

Kaliakatsos-Papakostas, M. (2014). Probabilistic harmonisation with fixed intermediate chord constraints. In *Proceedings of the 11th Sound and Music Computing Conference (SMC) and 40th International Computer Music Conference (ICMC)*, Athens, Greece.

Kirlin, P. B. (2009). Using harmonic and melodic analyses to automate the initial stages of Schenkerian analysis. In *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR 2009)*, pages 423–428, Kobe, Japan.

Kröger, P., Passos, A., and Sampaio, M. (2010). A survey of automated harmonic analysis techniques. Technical Report 1, Universidade Federal da Bahia.

Lidy, T., Rauber, A., Pertusa, A., and Iñesta, J. M. (2007). Improving genre classification by combination of audio and symbolic descriptors using a transcription system. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, pages 61–66, Vienna, Austria.

Margineantu, D. D. and Dietterich, T. G. (2003). Improved class probability estimates from decision tree models. In *Nonlinear Estimation and Classification*, pages 173–188. Springer.

Marsden, A. (2010). Schenkerian analysis by computer: A proof of concept. *Journal of New Music Research*, 39(3):269–289.

Martin, J. G. (1972). Rhythmic (hierarchical) versus serial structure in speech and other behavior. *Psychological Review*, 79(6):487–509.

Maxwell, H. J. (1984). *An artificial intelligence approach to computer-implemented analysis of harmony in tonal music*. PhD thesis, Indiana University.

Mearns, L. (2013). *The computational analysis of harmony in Western art music*. PhD thesis, School of Electronic Engineering and Computer Science, Queen Mary University of London.

Meredith, D. (1993). Computer-aided comparison of syntax systems in three piano pieces by Debussy. *Contemporary Music Review*, 9(1-2):285–304.

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

Mouton, R. and Pachet, F. (1995). The symbolic vs. numeric controversy in automatic analysis of music. In *Proceedings of the Workshop on Artificial Intelligence and Music (IJCAI 1995)*, pages 32–39.

Pachet, F. (1991). A meta-level architecture applied to the analysis of jazz chord sequences. In *Proceedings of the International Computer Music Conference*, pages 1–4, Montreal, Canada.

Pachet, F. and Roy, P. (2000). Musical harmonization with constraints: A survey. *Constraints*, 6(1):7–19.

Pardo, B. and Birmingham, W. (2002). Algorithms for chordal analysis. *Computer Music Journal*, 26(2):27–49.

Pardo, B. and Birmingham, W. P. (2000). Automated partitioning of tonal music. *FLAIRS Conference*, pages 23–27.

Passos, A., Sampaio, M., Kröger, P., and de Cidra, G. (2009). Functional harmonic analysis and computational musicology in Rameau. In *Proceedings of the 12th Brazilian Symposium on Computer Music (SBMC)*.

Pérez-García, T., Iñesta, J. M., Ponce de León, P. J., and Pertusa, A. (2011). A multimodal music transcription prototype. In *Proceedings of ACM International Conference on Multimodal Interaction (ICMI 2011)*, pages 315–318, Alicante, Spain.

Pérez-Sancho, C., Rizo, D., and Iñesta, J. M. (2009). Genre classification using chords and stochastic language models. *Connection Science*, 21(2,3):145–159.

Phon-Amnuaisuk, S., Smaill, A., and Wiggins, G. (2006). Chorale harmonization: A view from a search control perspective. *Journal of New Music Research*, 35(4):279–305.

Piston, W. (1987). *Harmony*. W. W. Norton & Company. Revised and expanded by Mark DeVoto.

Prather, R. (1996). Harmonic analysis from the computer representation of a musical score. *Communications of the ACM*, 39(12es):239–255.

Quinlan, J. R. (2014). *C4.5: Programs for Machine Learning*. Elsevier.

Raczyński, S., Fukayama, S., and Vincent, E. (2013). Melody harmonization with interpolated probabilistic models. *Journal of New Music Research*, 42(3):223–235.

Radicioni, D. and Esposito, R. (2007). Tonal harmony analysis: A supervised sequential learning approach. In Basili, R. and Pazienza, M. T., editors, *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*, volume 4733 of *Lecture Notes in Artificial Intelligence*, pages 638–649. Springer.

Rameau, J.-P. (1722). *Traité de l'harmonie, reduite à ses principes naturels*. Jean-Baptiste-Christophe Ballard.

Ramírez, R., Perez, A., Kersten, S., Rizo, D., Roman, P., and Iñesta, J. M. (2010). Modeling violin performances using inductive logic programming. *Intelligent Data Analysis*, 14(5):573–585.

Raphael, C. and Nichols, E. (2008). Training music sequence recognizers with linear dynamic programming. In *Proceedings of the International Workshop on Machine Learning and Music (MML)*, Helsinki, Finland.

Raphael, C. and Stoddard, J. (2004). Functional harmonic analysis using probabilistic models. *Computer Music Journal*, 28(3):45–52.

Rizo, D. (2010). *Symbolic music comparison with tree data structures*. PhD thesis, Universidad de Alicante.

Rohrmeier, M. (2007). A generative grammar approach to diatonic harmonic structure. In *Proceedings of the 4th Sound and Music Computing Conference*, pages 97–100.

Rohrmeier, M. (2011). Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*, 5(1):35–53.

Rothgeb, J. E. (1968). *Harmonizing the unfigured bass: A computational study*. PhD thesis, Yale University.

Sabater, J., Arcos, J., and López de Mántaras, R. (1998). Using rules to support case-based reasoning for harmonizing melodies. In Freuder, E., editor, *Multimodal Reasoning: Papers from the 1998 AAAI Spring Symposium (Technical Report, SS-98-04)*, pages 147–151, Menlo Park, CA. AAAI Press.

Sapp, C. (2007). Computational chord-root identification in symbolic musical data: Rationale, methods, and applications. *Computing in Musicology*, 15:99–119.

Sapp, C. (2011). *Computational methods for the analysis of musical structure*. PhD thesis, Stanford University.

Scarborough, D. L., Miller, B. O., and Jones, J. A. (1989). Connectionist models for tonal analysis. *Computer Music Journal*, 13(3):49.

Scholz, R., Dantas, V., and Ramalho, G. (2005). Automating functional harmonic analysis: The Funchal system. In *Seventh IEEE International Symposium on Multimedia*, Irvine, CA.

Simon, I., Morris, D., and Basu, S. (2008). MySong: Automatic accompaniment generation for vocal melodies. In *CHI '08: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 725–734, New York, NY.

Suzuki, S. and Kitahara, T. (2014). Four-part harmonization using Bayesian networks: Pros and cons of introducing chord nodes. *Journal of New Music Research*, 43(3):331–353.

Taube, H. (1999). Automatic tonal analysis: Toward the implementation of a music theory workbench. *Computer Music Journal*, 23(4):18–32.

Taube, H. and Burnson, W. A. (2008). Software for teaching music theory. Technical report, University of Illinois at Urbana-Champaign.

Temperley, D. (1997). An algorithm for harmonic analysis. *Music Perception*, 15(1):31–68.

Temperley, D. (2001). *The Cognition of Basic Musical Structures*. The MIT Press.

Temperley, D. (2004). Bayesian models of musical structure and cognition. *Musicae Scientiae*, 8(2):175–205.

Temperley, D. and Sleator, D. (1999). Modeling meter and harmony: A preference-rule approach. *Computer Music Journal*, 23(1):10–27.

Tojo, S., Oka, Y., and Nishida, M. (2006). Analysis of chord progression by HPSG. In *AIA'06: Proceedings of the 24th IASTED international conference on Artificial intelligence and applications*.

Toselli, A. H., Vidal, E., and Casacuberta, F. (2011). *Multimodal Interactive Pattern Recognition and Applications*. Springer.

Tracy, M. S. (2013). *Bach in Beta: Modeling Bach chorales with Markov Chains*. PhD thesis, Harvard University.

Tsui, W. (2002). Harmonic analysis using neural networks. Master's thesis, University of Toronto.

Ulrich, J. W. (1977). The analysis and synthesis of jazz by computer. In *Proceedings of the 5th International Joint Conference on Artificial intelligence (IJCAI)*.

Willingham, T. J. (2013). *The harmonic implications of the non-harmonic tones in the four-part chorales of Johann Sebastian Bach*. PhD thesis, Liberty University.

Winograd, T. (1968). Linguistics and the computer analysis of tonal harmony. *Journal of Music Theory*, 12(1):2–49.

Yi, L. and Goldsmith, J. (2007). Automatic generation of four-part harmony. In *Proceedings of the Fifth UAI Bayesian Modeling Applications Workshop (UAI-AW 2007, BMA'07*, Vancouver, Canada.