A Conditional Model for Tonal Analysis

Daniele P. Radicioni and Roberto Esposito

Università di Torino, Dipartimento di Informatica
C.so Svizzera 185, 10149, Turin, Italy
{radicion, esposito}@di.unito.it

Abstract. Tonal harmony analysis is arguably one of the most sophisticated tasks that musicians deal with. It combines general knowledge with contextual cues, being ingrained with both faceted and evolving objects, such as musical language, execution style, or even taste. In the present work we introduce BREVE, a system for tonal analysis. BREVE automatically learns to analyse music using the recently developed framework of conditional models. The system is presented and assessed on a corpus of Western classical pieces from the 18^{th} to the late 19^{th} Centuries repertoire. The results are discussed and interesting issues in modeling this problem are drawn.

1 Introduction

Music analysis is arguably one of the most sophisticated tasks that musicians deal with. It combines general knowledge with contextual cues, being ingrained with both faceted and evolving objects, such as musical language, execution style, and even taste. Besides, analysis is a crucial step in order to deal with music. Performers, listeners, musicologists, automatic systems for performance, composition and accompaniment: all of them need to process and "understand" the pieces being composed, listened or performed. Consequently, analysis has been addressed by didactic literature [1]; psychological concerns have inspired theories about listeners perception [2], as well as about performers strategies and constraints [3]. Such seminal studies have been exploited to the end of devising automatic expressive performers in the AI field (see, e.g., [4,5]).

Within the broader music analysis area, we single out the task of tonal harmony analysis. Analyzing music harmony consists of associating a label to each *vertical* (that is, set of simultaneous notes) [6,7]. Such labels explain which harmony is sounding by indicating a chord name in terms of a *root* (the fundamental note) and a *mode*, such as C minor (Fig. 1).

Tonal harmony theory encodes how to build chords (i.e., which pitches compose each chord) and how to concatenate them. This task is, in general, a difficult one: in fact, music students spend considerable amounts of time in learning tonal harmony, which is still the base of both classical composer's and performer's background. Moreover, we have to handle ambiguous cases [8], and composer's strategies aiming at surprising those of the listener, by suddenly introducing elements that violate expectation. In fact, to some extent musical language has

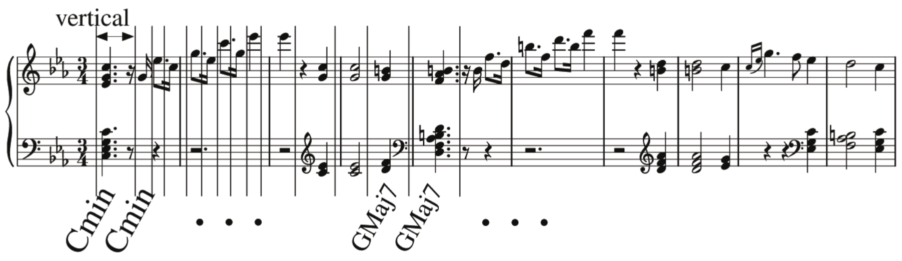


Fig. 1. The tonal harmony analysis problem consists of indicating for each vertical which chord is currently sounding. Excerpt from Beethoven's Piano Sonata Op.10 n.1.

evolved under such a pressure for breaking "grammatical" rules [9]. This heightens the actual complexity of analysis, and also the difficulties encountered by automatic analysis systems.

It is worth mentioning two subtle points. Firstly, music *analysis* and *composition* tasks are conceptually different, though related. Analyzing music is introductory to composing music, and it plays a central role in systems for composition (see, e.g., the pioneering system by Ebcioglu [10]). In the present work, we do not describe the next artificial composer, but rather an analysis system. Secondly, an higher level of analysis (*functional* analysis) exists that aims at individuating harmonic functions in chords [9]. As it will be clearer later on, this kind of analysis is a long term goal for harmony analysis systems. In the present work, we address the basic form of harmonic analysis mentioned earlier, and similar to [6] and [7].

In summary, music analysis is a complex problem, requiring many hours of apprenticeship to professional musicians. This makes such knowledge difficult to be elicited, and enhances the difficulties encountered while devising systems to accomplish automatically the task. The present system attempts to tackle the problem with machine learning techniques, which also benefit from an encoding of harmony theory knowledge.

2 Related Works on Tonal Harmony Analysis

Much work has been carried out in the field of automatic tonal analysis: since the pioneering grammar-based work by Winograd [11], a number of approaches have been proposed that address the issue of tonal analysis.

One of the preference rules systems described by Temperley [6] is devoted to harmonic structure analysis. It relies on the Generative Theory of Tonal Music [2], providing that theory with a working implementation. In this approach, preference rules are used to evaluate possible kinds of analysis, such as harmony analysis, also exploiting meter, grouping, and pitch spelling information. A major feature of Temperley's work is in applying high level domain knowledge, such as "Prefer roots that are close to the roots of nearby segments on the line of fifths", which leads to an explanation of results.

The system by Pardo & Birmingham [7] is a template matching system. It performs tonal analysis by assigning a score to a set of 72 templates (from the combination of 6 original templates transposed over 12 semi-tones of the chromatic scale). The resulting analysis is further improved by 3 tie-resolution rules, for labeling still ambiguous cases.

Raphael & Stoddard [12] propose a machine learning approach based on a Hidden Markov Model that computes roman numeral analysis (that is, the functional analysis mentioned above). A main feature of their system is that the generative model can be trained using unlabeled data, thus determining its applicability also to unsupervised problems. In order to reduce the huge number of parameters to be estimated, they make a number of assumptions, such as that the current chord does not affect the *key transitions*. Also, the generative model assumes conditional independence of notes (the observable variables) given the current mode/chord.

Our approach relies on the recent paradigm of *conditional models* [13]. Conditional models directly solve the conditional problem of predicting the label given the observations, instead of modeling the joint probability of observations and labels. We argue that –in the specific case of music *analysis*– the generative features of HMMs are neither required nor useful, as this actually results in forcing a more complex model to be estimated, without any additional benefit with respect to conditional models.

Unfortunately, experimental comparisons between harmony analysis systems are not that simple. The system by Temperley does not consider the mode of the chord (thus resulting in a different kind of analysis), while the algorithm used by Pardo is deeply affected by the accuracy of the segmentation. On the other hand, based on methodological accounts, our system should be compared with that from Raphael & Stoddard, although they do no provide results of a systematic experimentation.

3 The System

The chief idea behind our work is that music analysis task can be naturally cast to a Machine Learning (ML) problem, suitable to be solved by Supervised Sequential Learning (SSL) techniques. In fact, by considering only the "vertical" aspects of musical structure, one would hardly produce reasonable analyses. As mentioned in [14], experimental evidences about human cognition reveal that composers and listeners refer to "horizontal" features of music to disambiguate unclear cases. Therefore context plays a fundamental role and contextual cues are deemed useful to the analysis system. Moreover, harmony changes are well known to follow patterns implying that analysis must take into consideration the succession of chords (e.g., the case of *cadence*).

The fundamental representational structure processed by BREVE is the music *event*; whole pieces are represented as *event lists*. An event is a set of *pitch*

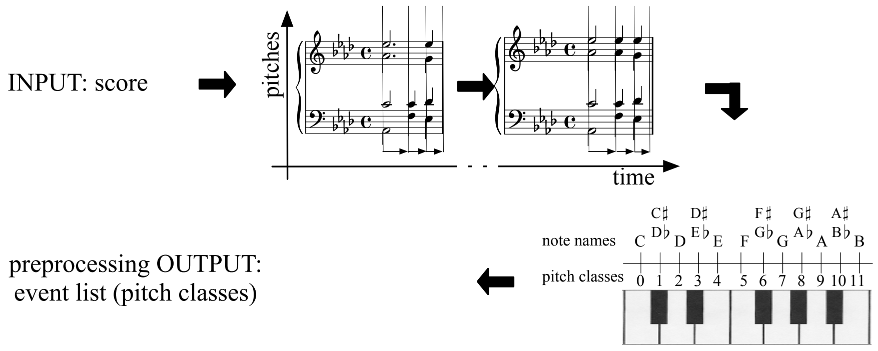

INPUT: score
pitches
time
preprocessing OUTPUT: event list (pitch classes)
note names
C C♯ D♭ D D♯ E♭ E F F♯ G♭ G G♯ A♭ A A♯ B♭ B
pitch classes
0 1 2 3 4 5 6 7 8 9 10 11

Fig. 2. Top: passing from the "Common Practice Notation", displayed on the left side, to the corresponding "explicit" representation, on the right side. Bottom: notes are then converted into the corresponding pitch classes.

classes sounding at the same time. Any note onset or offset determines a new event. Pitch classes are categories "such that any two pitches one or more octaves apart are members of the same category" [6]. In other words, pitch classes are congruence classes under the modulo-12 relation. For example, a vertical composed by the notes C4-E4-G4, corresponding to the MIDI pitch numbers 60-64-67, is converted into an event composed by the pitch classes 0-4-7. We retain information about *duration* of events as well. For example, if a note i is held while a new note j is played, we consider an event containing i, and an event composed by both i and j since the held note affects the harmonic content of the new vertical as well (top of Figure 2).

Much of the information required to perform tonal harmony analysis lies in the notes currently sounding and in their metrical salience: specifically, accented events are expected to convey much harmonic content [2]. Hence for each event we not only encode which pitch classes are present, but also whether the event is accented or not. Meter estimation is based on the work of Temperley [6].

3.1 Learning Algorithm

To analyze the harmonic structure of a piece is a sequential task, where contextual cues are commonly acknowledged to play a fundamental role [2]. It follows that standard classification approaches such as decision trees, naive bayes, etc. are arguably not likely to produce good classification hypotheses, since they do not take into account the contextual information. This information is provided by surrounding observations, as well as nearby labeling.

The learning task consists in seeking an hypothesis H which accurately predicts the labels associated to events. The task, known as the Supervised Sequential Learning (SSL) task, is not novel to the Machine Learning community. A wealth of research has been invested to develop algorithms for solving this kind of problems, and a number of interesting algorithms have been proposed [15,13]. The SSL task can be specified as follows [16]:

Given: A set L of training examples of the form (X_m, Y_m), where each $X_m = (x_{m,1}, \ldots, x_{m,T_m})$ is a sequence of T_m feature vectors and each $Y_m = (y_{m,1}, \ldots, y_{m,T_m})$ is a corresponding sequence of class labels, $y \in \{1, ..., K\}$.
Find: A classifier H that, given a new sequence X of feature vectors, predicts the corresponding sequence of class labels $Y = H(X)$ accurately.

In our case, each X_m corresponds to a particular piece of music; $x_{m,t}$ is the information associated to the event at time t; and $y_{m,t}$ corresponds to the chord label (i.e., the chord root and mode) associated to the event sounding at time t. The problem is, thus, to learn how to predict accurately the chord labels given the musical events information. In case the specification of the identity of an example is not relevant, we drop the m subscript in the X_m notation. In such situations, an event at time t within a sequence X is denoted as $x_{\cdot,t}$.

The SSL problem can be solved with several techniques, such as Sliding Windows, Hidden Markov Models, Maximum Entropy Markov Models [17], Conditional Random Fields [18], and Collin's adaptation of the Perceptron algorithm to sequencial problems [13] (henceforth, HMPerceptron). All these methods, with the exception of Sliding Windows, are generalizations and improvements of Markovian models, culminating in Conditional Random Fields and HMPerceptrons. Conditional Random Fields (CRFs) are state of the art conditional probabilistic sequence models, that improve on Maximum Entropy models [18,15] while maintaining most of the good properties of conditional models.

Unfortunately, the parameter estimation algorithm given for CRFs is known to be slow, in particular when a large number of features are to be handled [19]. A faster and simpler approach in learning conditional models has been recently proposed by Collins [13]. The algorithm, which is an extension to sequential problems of Rosenblatt's Perceptron algorithm, is reportedly at least on par with Maximum Entropy and CRFs models from the point of view of classification accuracy. To the best of our knowledge, a direct comparison of HMPerceptron and CRFs has not been provided, even though they both were applied to the same Part-Of-Speech tagging problem, with similar results [13,18].

Therefore, on the basis of cited literature, we have chosen the HMPerceptron as our main learning algorithm for the harmonic labeling prediction task. The hypothesis acquired by the HMPerceptron has the form:

$$H(X) = \arg \max_{\overline{Y}=\{\overline{y}_1 \cdots \overline{y}_T\}} \sum_t \sum_s w_s \phi_s(X, \overline{y}_t, \overline{y}_{t-1}) \tag{1}$$

where ϕ_s is a *boolean* function of the sequence of events X and of the previous and current labels. The ϕ_s functions are called "features" and allow the algorithm to take into consideration different aspects of the sequence being analyzed. To devise properly the set of features is the "difficult" part in applying the model, since this is the place where both the domain knowledge and the model "simplifications" come to play. The w_s weights are the parameters that need to be estimated by the learning algorithm. The HMPerceptron applies a simple scheme to do that: it iterates over the training set updating the weights so that features correlated to "correct" outputs receive larger values, and those correlated with "incorrect" ones receive smaller values.

This is actually the same kind of strategy adopted by Rosemblat's perceptron algorithm [20], the only real difference between the two algorithms is in the way the hypothesis is evaluated. In the classification problem faced by the perceptron algorithm, in fact, it is sufficient to enumerate all the possible labels and to pick the best one. In the case of the SSL problem, however, this cannot be done efficiently: the labelling of a single "example" is a sequence of T labels, the number of such labellings is thus exponential in T. To overcome this problem, the algorithm uses Viterbi decoding to pick the best labelling under a first order Markov assumption.

Features Definition. The features we use to model the tonal harmony problem can be arranged into the following classes: *Notes*, *Chord transitions*, *Metric relevance*, *Chord labels* and *Template matching.* In general, features are used to provide discriminative power to the learning system. At each time point t, the features are evaluated in order to compute an informed prediction about the label to associate with the event. The formal definition of the features exploited by the system is provided in Table 1.

Notes features report about the presence/absence of each one of the 12 available pitch classes in each event. For example, the occurrence in the training set of the pitches $\langle 9, 0, 4\rangle$ (that is $\langle$A,C,E$\rangle$) associated with the label "A min", provides evidence that these three pitch classes can be appropriately used to strengthen "A min" label for the event at time t. This information is provided for times t, $t{-}1$ and $t{+}1$. So we allow the algorithm to make use of previous and following notes to discriminate among possible chord labels.

Chord transitions feature reports about transitions between chord pairs. The feature allows the HMPerceptron to take into consideration the previous chord label in making the chord prediction about the current event.

Metric relevance features account for the correlation of label changes and the *beat level* of the event being analyzed and of nearby events. In general, the way the meter changes in the neighborhood of a given event is reputed relevant in predicting harmony changes [2]. Then, the metric relevance feature has been devised so to take into consideration the metrical salience of the current and of nearby events as well as the changes in harmony. Since we consider only two possible levels of metric relevance (corresponding to accented and unaccented beats) and adjacent neighbors (at times t–1 and t+1), we end up with eight possible metrical patterns.

Chord label features report about which label is actually asserted. Intuitively, these features allow the algorithm to evaluate something similar to prior probabilities of the various labels. Here one should consider that there are many possible labels. These features have been devised with the aim at biasing the analysis towards most frequent labels. In principle, these features should be used, other things being equal, to favor frequently stated labels.

Table 1. Feature's formal definition. Characteristics of the feature vector $x_{\cdot,\cdot}$ are denoted by characteristic-name$[x_{\cdot,\cdot}]$. Three "characteristics" are used in computing the features: "notes" reports the set of pitch classes corresponding to the notes in $x_{\cdot,\cdot}$; "meter" is 1 if event $x_{\cdot,\cdot}$ is accented and 0 otherwise; "tm-chord" reports the chord inferred by the template matching algorithm. Each definition reports the condition that needs to hold for the feature to return 1. Also, each feature reported in the table is actually a feature template which depends on several parameters. The third column clarifies the role of such parameters in each definition. $\overline{y}_t$ and $\overline{y}_{t-1}$ are defined as in Formula (1).

Name	Definition	Parameters
notes	$z \in \text{notes}[x_{\cdot,t+i}] \wedge \overline{y}_t = y'$	z: a pitch class; i: one of -1,0,1 y': a chord label
chords	$\overline{y}_t = y'$	y': a chord label
chords transitions	$\overline{y}_{t-1} = y'_1 \wedge \overline{y}_t = y'_2$	y'_1, y'_2: chord labels
metric relevance	$\text{meter}[x_{\cdot,t-1}] = m_1 \wedge \text{meter}[x_{\cdot,t}] = m_2 \wedge \text{meter}[x_{\cdot,t+1}] = m_3 \wedge \overline{y}_{t-1} = \overline{y}_t$	m_1, m_2, m_3: 1/0
template matching	$\text{tm-chord}_i[x_{\cdot,t}] = \overline{y}_t$	i: template index

Template matching features report about the chords predicted by a simple template matching algorithm that matches the event against a number of "standard" harmony templates. These features formulate up to five hypotheses for each vertical, abstaining when unsure.

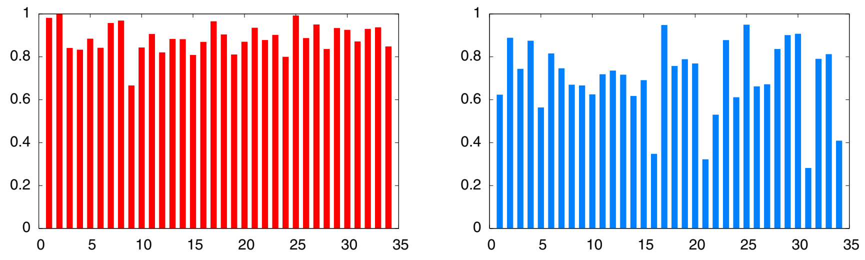
4 Experimental Validation

In order to assess our system, we adopted the Kostka-Payne corpus [8], which is a collection of 45 excerpts from the classical tonal repertoire, ranging from J.S. Bach (1685-1750) to N.D. Ayer (1910-1989), from piano pieces to string Trios and Quartets. B. Pardo annotated the analyses of Kostka and Payne into a set of MIDI files, and made the corpus available on the Net[1]. Unfortunately, we were not able to use the full corpus for the experimentation: 11 excerpts could not be used because of problems we encountered in parsing the MIDI files[2]. The final dataset contains a total of 2,622 events in 34 different sequences for an average of 77 events per sequence and a standard deviation of 41 events. The minimum and maximum number of events per sequence are 24 and 208 respectively. There are 73 different chord labels in the whole data set.

Figure 3 reports the performances of two experiments. The first experiment (Figure 3(a)) reports about the average errors made on each musical piece of the corpus. The performances of the system on the training set are reasonably good, they reach an average accuracy of 89.05%, thus demonstrating that the discrimination power of the system is adequate for the problem at hand.

[1] `http://www.cs.northwestern.edu/~pardo`

[2] We used the parsing routines provided by the standard javax.sound.midi package. The full list of the excerpts in the KP corpus is provided in [21].



(a) Accuracies on the training set (b) Leave-one-out accuracies

Fig. 3. Accuracies of the BREVE system on the KP corpus

In the second experiment (Figure 3(b)), the system has been iterated 34 times; each time, one of the excerpts has been removed from the training set, and used for testing. This is a leave-one-out cross validation setup, considering each piece one example. The final accuracy of the system is 73.79%.

4.1 Discussion

Looking at the results, we notice that they are accurate but also they overfit the data. This is somewhat surprising, because we were careful in implementing the averaging parameters method suggested in [13]. As reported in that paper, it should guarantee good generalization performances by making the HMPerceptron work similarly to a very simple ensemble learning algorithm. A possible explanation is that the selected corpus is very challenging from a machine learning perspective. In fact, it encompasses excerpts from authors of very different epochs and styles: some of these appear only once, thus implying that the system is required to generalize from observations encountered in very different excepts. Also, the above remarks seems to be confirmed by the results of a very similar system that we tested on a much more homogeneous corpus composed only by J.S. Bach chorales: in that case the system achieved about 90% of generalization accuracy [14].

A closer look at our results reveals that most of the errors can be arranged in two main classes, thus providing precious insights to improve the system. A first, obvious, remark is that the sequential information is harder to capture than the information present in the individual events. For instance, there are cases in which the perceptron may have been induced to return such sequence as a result of having being trained over a corpus where romantic language is more represented[3]. In fact, such cases point out a romantic nuance in the interpretation, that could be corrected should a larger data set be available. A possible way to surmount the problem is to provide the system with some further domain knowledge by adding higher level features. Several errors (namely, 16.34%) are

[3] E.g., cases where the sequences have been interpreted in terms of III-II$^{\varnothing}$-i instead of III-VI-i.

due to the confusion between relative tones: i.e., our output was a major chord, while the correct answer was its relative minor, or vice versa. In a sense, this is itself an encouraging error, indicating that the system provides reasonable output. In the 17.32% of the overall error, we correctly identified the root but not the mode. We registered such problems especially when dealing with arpeggios (e.g., the first movement of the "Moonlight" Sonata Op. 27 n.1 by Beethoven), where harmony is incompletely stated, and it goes back and forth between C♯ minor and major. In both cases, it would have been crucial to consider features that take into account larger contexts. In several cases, our analysis is somehow more concise (or less detailed) than the "correct" one, disregarding, for instance, passing tones. Sometimes we noted the opposite problem, i.e., BREVE provides analyses that are more detailed. Again, collecting information from a broader neighborhood is likely to help much in those situations. The good news are that enlarging the context should not affect very much the performances. In fact conditional models allow considering arbitrarily distant contextual cues, without high additional costs, as long as they do not require the evaluation of faraway labels.

5 Conclusions

This paper has presented BREVE, a novel system for performing tonal harmony analysis. The main contributions are the following. We developed a system where a HMPerceptron exploits pitch classes in nearby events, metrical accents patterns, chordal successions, as well as suggestions given by a template matching algorithm. The system has been subject to a systematic experimentation and provided encouraging results. Most errors fall into few classes. Their analysis provides meaningful insights about the system behavior.

The analysis problem confirmed to be both interesting and challenging. Part of the difficulties surely comes from the heterogeneity and meagerness of the selected excerpts corpus, but the evidence is that much work is still needed to build systems suitable for tackling the problem in its entireness. Problems come from different directions: known algorithms still require large computational resources; higher level features need to be engineered to improve performances and accuracy; knowledge extraction tools, still beyond the reach of actual systems, would help scientists to explain how music is composed, listened, performed and, ultimately, understood. To develop such higher level features, to augment the corpus of the data, to extract declarative knowledge from the inferred hypotheses, are all interesting directions for future research.

References

1. Bent, I., ed.: Music Analysis in the Nineteenth Century. Cambridge University Press, Cambridge (1994)
2. Lerdahl, F., Jackendoff, R.: A Generative Theory of Tonal Music. MIT Press, Cambridge, MA (1983)

3. Sloboda, J.: The Musical Mind. The Cognitive Psychology of Music. Oxford University Press, Oxford, UK (1985)
4. Widmer, G.: On the Potential of Machine Learning for Music Research. In: Readings in Music and Artificial Intelligence. Harwood Academic Publishers (2000)
5. Lopes de Mantaras, R., Arcos, J.: AI and Music: From Composition to Expressive Performance. AI Magazine **23** (2002) 43–57
6. Temperley, D.: The Cognition of Basic Musical Structures. MIT, Cambridge, MASS (2001)
7. Pardo, B., Birmingham, W.P.: Algorithms for Chordal Analysis. Computer Music Journal **26** (2002) 27–49
8. Kostka, S., Payne, D.: Tonal Harmony. Number New York. McGraw-Hill (1984)
9. Schoenberg, A.: Structural Functions in Harmony. Norton, New York (1969)
10. Ebcioglu, K.: An Expert System for Harmonizing Chorales in the Style of J. S. Bach. Journal of Logic Programing **8**(1) (1990) 145–185
11. Winograd, T.: Linguistics and Computer Analysis of Tonal Harmony. Journal of Music Theory **12** (1968) 2–49
12. Raphael, C., Stoddard, J.: Functional Harmonic Analysis Using Probabilistic Models. Computer Music Journal **28**(3) (2004) 45–52
13. Collins, M.: Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. (2002)
14. Radicioni, D.P., Esposito, R.: Learning Tonal Harmony from Bach Chorales. In Fum, D., del Missier, F., Stocco, A., eds.: Procs. of the 7th International Conference on Cognitive Modelling. (2006)
15. Dietterich, T. In: Machine Learning for Sequential Data: A Review. Volume 2396 of Lecture Notes in Computer Science. Springer-Verlag (2002) 15–30
16. Dietterich, T.G., Ashenfelter, A., Bulatov, Y.: Training Conditional Random Fields via Gradient Tree Boosting. In: Procs. of the 21st International Conference on Machine Learning, New York, NY, USA, ACM Press (2004)
17. McCallum, A., Freitag, D., Pereira, F.: Maximum Entropy Markov Models for Information Extraction and Segmentation. In: Procs. of the 17th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA (2000) 591–598
18. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Procs. of the 18st International Conference on Machine Learning, San Francisco, CA (2001) 282–289
19. Altun, Y., Hofmann, T., Smola, A.J.: Gaussian Process Classification for Segmenting and Annotating Sequences. In Greiner, R., Schuurmans, D., eds.: Proceedings of the Twenty-first International Conference on Machine Learning, Banff, Canada, ACM Press, New York, NY (2004)
20. Rosenblatt, F.: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review (Reprinted in Neurocomputing (MIT Press, 1998)) **65** (1958) 386–408
21. Radicioni, D.P., Esposito, R.: Sequential Learning for Tonal Analysis. Technical Report RT 92/05, Università degli Studi di Torino (December 2005)