LEARNING JAZZ GRAMMARS

Jon Gillick
Wesleyan University
Middletown, Connecticut, USA
jrgillick@wesleyan.edu

Kevin Tang
Cornell University
Ithaca, New York, USA
kt258@cornell.edu

Robert M. Keller
Harvey Mudd Collge
Claremont, California, USA
keller@cs.hmc.edu

ABSTRACT

We are interested in educational software tools that can generate novel jazz solos in a style representative of a body of performed work, such as solos by a specific artist. Our approach is to provide automated learning of a grammar from a corpus of performances. Use of a grammar is robust, in that it can provide generation of solos over novel chord changes, as well as ones used in the learning process. Automation is desired because manual creation of a grammar in a particular playing style is a labor-intensive, trial-and-error, process.

Our approach is based on unsupervised learning of a grammar from a corpus of one or more performances, using a combination of clustering and Markov chains. We first define the basic building blocks for contours of typical jazz solos, which we call "slopes", then show how these slopes may be incorporated into a grammar wherein the notes are chosen according to tonal categories relevant to jazz playing. We show that melodic contours can be accurately portrayed using slopes learned from a corpus. By reducing turn-around time for grammar creation, our method provides new flexibility for experimentation with improvisational styles. Initial experimental results are reported.

1. INTRODUCTION

Jazz improvisation is a form of composition done concurrently with the performance of the music itself. Although the ideal would have no premeditation about what will be performed, it is known that jazz musicians do work out and practice vocabulary ideas prior to the actual performance. We are interested in tools that facilitate the construction and recording of such ideas, for purposes of education as well as performance. The present contribution demonstrates that grammars for generating jazz melodies can be learned from performances, in a manner that captures stylistic aspects of the performer. The ability to generate melodies in a given style is expected to have significant tutorial value.

2. RELATED WORK

Use of grammars for creating musical structures has been investigated by Cope [6], Bel [4], Pachet [18], and others. We base our approach on the grammatical representation of Keller and Morrison [14], which seems to provide an adequate basis for generating jazz melodies.

Methods for algorithmic composition have been surveyed extensively by McCormack [17] and by Papadopoulos and Wiggins [19]. Our work combines and extends some of the ideas they discussed, including the combined use of grammars and machine learning. Dubnov, Assayag, Lartillot, and Bejerano [8] used probabilistic and statistical machine learning methods for musical style recognition. Eck and Lapamle [9] investigated automatic composition and improvisation with neural networks, and Cruz-Alcazar and Vidal-Ruiz [1] developed a method for learning grammars to model musical style.

The idea of melodic contour for abstraction has been used for analysis purposes. Kim, Chai, Garcia and Vercoe [15] used contours for musical classification and querying, and Chang and Jiau [5] investigated musical contour with applications to extracting repeating figures and themes from music. In addition, De Roure and Blackburn [7] proposed melodic pitch contours for content-based navigation of music. We incorporate the melodic abstraction of contours and slopes by utilizing them as the building blocks of a grammar for compositional purposes.

Kang, Ku, and Kim [13] used a graphical clustering algorithm for extraction of melodic themes. The clustering portion of our model serves a similar function. Verbeurgt, Dinolfo, and Fayer [20], among others, used Markov models as a means for composition by learning transition probabilities between patterns. Ames [2] dealt with different-sized Markov chains of notes. Our approach utilizes Markov chains to determine transition probabilities between *clusters* of different types of melodic themes. We believe this is helpful in providing additional flexibility and fluidity needed to generate jazz melodies.

3. ABSTRACT MELODIES

Although a given jazz performer might not be aware of how he or she does improvise, it seems reasonable to say that ideas of what one is able and willing to play can be captured in some form of grammar. At the very least, if a student has memorized a finite set of "licks" (melodic fragments), it is obvious that that set could be described by an *ad hoc* grammar. A grammar that is too *ad hoc*, however, would tend to generate only very predictable, and thus eventually uninteresting, melodies. It is important then that melodic ideas be abstracted so as to enable the replacement of certain elements with others to continually produce novel output. If the abstraction is too coarse-grain, though, the melody may lose coherence.

Our grammatical approach for jazz melodies attempts to strike a balance between novelty and coherence by augmenting the five note categories of [14] that correspond to concepts in jazz playing. These categories are instantiated probabilistically and also in observance of other constraints, such as range considerations, at generation time. Each category, as given in Table 1, has a corresponding terminal symbol in the grammar and four of them show as different note head colors on the staff, for explication purposes.

Symbol	Color	Meaning
C	black	Chord tones of the current **Chord**
L	green	**Color tones,** complementary tones sonorous with the current chord
A	blue	Tones that chromatically **approach** one of the above
——	red	Neither C, L, nor A
S	——	Tones in a **scale** that corresponds to the chord
X	——	Arbitrary tone
R	——	Rest

Table 1. Note categories used in grammar terminals

A terminal symbol of the grammar is formed by following a category symbol by a numeric duration. For example, A8 represents an approach tone of duration one eighth-note, C4 a chord tone of duration one quarter-note, L4/3 a color tone of a quarter-note *triplet*, S4. a *dotted* quarter-note, R2 a half-note rest, etc. We think of a sequence of terminal symbols in the grammar as being an *abstract melody*, in the sense that multiple melodies will fit the sequence when the categories are instantiated to corresponding tones. Another advantage of such melodic abstractions is that they can be instantiated over any chord progression, even for chords of different quality, such as major vs. minor.

We extend these individual note categories with "macro" concepts dealing with sequences of notes in certain patterns. Although more general macros are possible, our current work focuses on a single macro concept, called a *slope*. Each slope has two numeric parameters, indicating the minimum and maximum interval between notes in the sequence going in the ascending direction. Negative numbers indicate the *descending* direction. S-expressions [16] are used to provide grouping of notes in a sequence, and for hierarchy, when necessary. For brevity, we will represent "slope" by Δ in this paper. For example, (Δ 1 2 S8 S8 S8) would indicate an ascending group of three eighth notes that are scale tones, with each at least 1 semitone and at most 2 semitones pitch separation. Similarly, (Δ -3 -4 C4 L4 C4) indicates a descending series of a chord, color, and chord tone, with a minimum separation of 3 and a maximum separation of 4 semitones. More generally, it is not always possible to obey the constraints of both the slope and the note category, so sometimes we must *relax* one or the other, as described later.

Slopes may be concatenated to provide *contours*. Each note symbol in a slope indicates a direction and a range of possibilities for the interval from the previous note. We break melodic lines into strictly ascending, descending, or stationary segments and define the slope of a segment by the minimum and maximum intervals between notes in the segment. Such a definition of slope allows us to represent many common jazz idioms. For example, consider the bebop idiom of an *enclosure* [3], wherein a chord tone is approached by notes above and below.

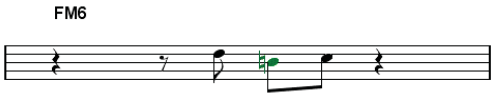


Figure 1. Example of an *enclosure*

Using slopes, we represent an abstraction of the melody in Figure 1 as the S-expression:

(R4 R8 L8 (Δ -3 -4 S8) (Δ 1 2 C8) R4).

Following two rests, we have an eighth note color tone followed by a scale tone three to four half steps down, a chord tone one to two steps up, and finally a quarter note rest. Note that we abstract only pitches, while rhythms are captured exactly.

Our notation for chords follows jazz lead sheet abbreviations, as given in Table 2.

Symbol	Meaning
M	major
m	minor
7	dominant seventh, if by itself
6	added sixth

Table 2. Jazz chord symbols used above the staff

In addition to short idioms, we can capture larger selections such as the line in Figure 2 from Red Garland's

solo on "Bye Bye Blackbird" [10]. We represent an abstraction of the melody in Figure 2 with another S-expression:

(R8 C8 (Δ -9 -9 A16)
(Δ 1 3 C16 C16 C16 C8)
(Δ -12 -12 C8)
(Δ 1 4 C8 A8)
(Δ -4 -1 L8 C8 C8 A8 C8)
(Δ 12 12 C8)
(Δ -12 -2 C8 C8))

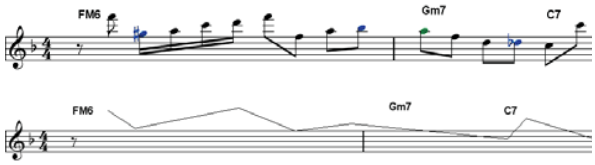
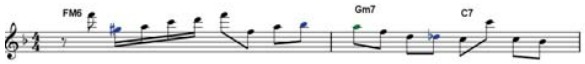


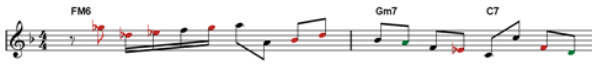
Figure 2. A melody line and its *slope* representation

Notes such as the G# in the first measure begin an ascending segment and so have only one interval from which to choose a minimum and maximum slope. In such cases, we found that relaxing the bounds by a half step in each direction yielded better results. Consequently, we relax (Δ -9 -9 A16) to (Δ -8 -10 A16) before instantiating to a melody. Since chord tones play the most significant role in shaping the melody, we weight chord tones higher than slope bounds, but for note categories other than chord tones, we do not.

Figure 3 demonstrates several new licks generated from our representation of Red Garland's melody.

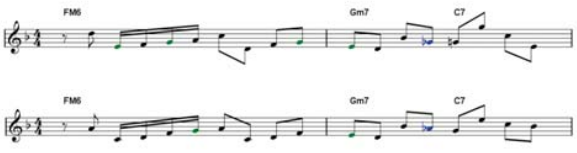


(a) Original melody



(b) Generation using only contours

(c) Generation using only note categories



(d) Two generations using both contour and categories

Figure 3. Original melody vs. melody generation methods

4. GRAMMATICAL INFERENCE

Grammatical inference (GI) algorithms attempt to define the rules of a grammar for an unknown language through analysis of a training dataset. The data can contain both positive sample sets (strings in the language) and negative sample sets (strings that should not be accepted), though we use only positive sample sets, *e.g.* performances from an artist that we want to imitate.

The idea of grammatical inference is to determine a set of rules that will produce strings similar to those in the training data. The grammar, then, should generate a set of strings containing everything in the training data and nothing that differs greatly from the data.

Cruz-Alcazar and Vidal-Ruiz [1] applied three GI algorithms to automatic composition of melodies in Gregorian, Bach, and Joplin styles, achieving the best results with the Gregorian melodies, in which they classified twenty percent of composed melodies as *very good*, which they define as able to be "taken as an original piece from the current style without being a copy or containing evident fragments from samples." We strove toward the same definition of *very good* solos.

We experimented with several methods for extracting grammar rules from training data, including extraction by phrases, with a phrase defined as a section of a solo starting after a rest and ending with a rest. Given the variable length of phrases and the difficulty of recombining phrases into a solo of a specified size, we settled on breaking melodies into *time windows* of a predefined length. After choosing two parameters, the number of beats per window and the number of beats by which to slide the window, we collect all melodic fragments of a certain length in a corpus and associate one grammar terminal for each abstract melody of the given length. We found that, among fragments of between 1 and 8 beats, 4 beat fragments achieved the best balance between originality and continuity. All songs in our training data are in 4/4 time, so we are effectively collecting abstractions for each measure in the training data.

5. MARKOV CHAINS

Once we have gathered the abstract melodies that will make up our generated solos, we combine them into full solos by implementing the equivalent of a Markov chain into the grammar. Markov chains represent a system with a sequence of states, using conditional probabilities to model the transitions between successive states. An n-gram Markov chain uses probabilities conditioned on the previous n-1 states. In our work, sets of abstract melodies serve as the states in the Markov chain; given a starting melody, we add the next phrase based on a list of transition probabilities from the first measure.

6. CLUSTERING

To construct a Markov chain with meaningful transition probabilities, we need a reasonable number of data points for each state. Before building the transition matrix, we group similar abstract measures together using the k-means clustering algorithm [11]. We then collect statistics on which clusters follow other clusters in the corpus and build our table of probabilities accordingly, using clusters as states of the Markov chain. To compose new solos, we first generate a sequence of clusters from the grammar and then randomly select representatives from clusters.

Clustering algorithms represent data as points in an n-dimensional plane and group points together through some distance metric. We base our cluster analysis as a Euclidean distance measure on 7 parameters:

1) Number of notes
2) Location of the first note struck within the window
3) Total duration of rests
4) Average maximum slope of ascending or descending groups of notes
5) Whether the window starts on or off the beat
6) Order of the contour (how many times it changes direction)
7) Consonance

We assign a "consonance" value to a measure based on the note categories. For each note, we add to the consonance value a coefficient for the note category multiplied by the length of the note. For example, typical coefficients are 0.8 for a chord note, 0.6 for an approach note, 0.4 for a color note, and 0.1 for other notes.

Given a parameter *k* for the number of clusters, we use the *k-means algorithm*, which selects *k* points as cluster centers and then begins an iterative process given by the following two steps:

1) Assign each data point to the nearest cluster center.
2) Re-compute the new cluster centers.

These steps are repeated for some number of iterations or until few enough data points switch clusters between iterations. Figure 4 shows three representative 1-measure melodies that the algorithm clustered together in a corpus of Charlie Parker solos.

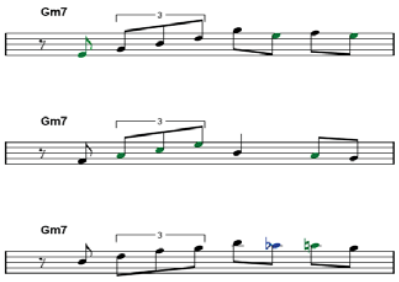


Figure 4. Three representatives from the same cluster

The top line of Figure 5 shows a 2-measure melody and the result of choosing two measures, one from the cluster for the first measure, and one for the cluster for the second.

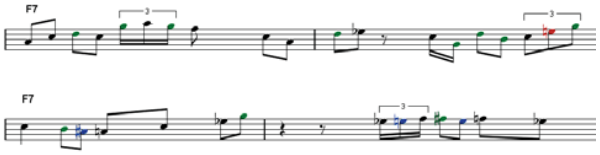


Figure 5. A 2-measure melody (top) and a melody synthesized from clusters corresponding to each of the measures (bottom)

7. IMPLEMENTATION

Our ideas and methods were implemented as a learning extension of the solo generation functionality of the open-source software tool Impro-Visor [12], implemented in Java. Both the executable and source code for the results presented here will be made available publicly.

Table 3 shows a simple grammar with 3 clusters inferred from a corpus of Charlie Parker solos. Each non-terminal symbol has an integer argument indicating the number of measures to be filled. Non-terminals C0, C1, and C2 represent the states of the Markov chain. Due to space limitations, we have only included one of the rules for each of Q0, Q1, and Q2. We have transcribed our implementation's notation from S-expressions to more conventional grammar rules for readability.

Production Rule	π
P(0) → ()	1
P(Y) → Start(32) P(Y-32)	1
Start(Z)→ C0(Z)	0.23
Start(Z) → C1(Z)	0.25
Start(Z) → C2(Z)	0.52
C0(0) → ()	1
C1(0) → ()	1
C2(0) → ()	1
C0(Z) → Q0 C0(Z-1)	0.24
C0(Z) → Q0 C1(Z-1)	0.24
C0(Z) → Q0 C2(Z-1)	0.52
C1(Z) → Q1 C0(Z-1)	0.18
C1(Z) → Q1 C1(Z-1)	0.28
C1(Z) → Q1 C2(Z-1)	0.54
C2(Z) → Q2 C0(Z-1)	0.25
C2(Z) → Q2 C1(Z-1)	0.24
C2(Z) → Q2 C2(Z-1)	0.51
Q0→((Δ 0 0 R2 R4 R8 C16/3)(Δ1 1 A16/3 L16/3)	0.33
Q1→((Δ 0 0 C8)(Δ -9 -9 C8)(Δ 2 3 C8 G4+8 R4))	0.33
Q2→((Δ 0 0 C4/3)(Δ 1 2 L4/3 A4/3)(Δ-7 -1 C4/3 G4 C8/3))	0.33

Table 3. Probabilistic grammar embedding a Markov chain, with π being the probability of using the rule, given the left-hand side

For brevity, the rules for expanding the individual clusters are not shown. These expand into sequences of slope specifications of the sort that were described in section 3. This grammar generates 32-measure abstract melodies, but we can specify any number of measures to generate, and the grammar will adjust accordingly. Once we have an abstract melody, we then generate a real melody by randomly selecting an initial note of the specified note category and then filling in the rest from slope and note category constraints. Figures 7 and 8 show two solos created by our approach, intended to be in the style of John Coltrane and Charlie Parker respectively. Each is based on a grammar learned from a corpus of solos from the respective artists.

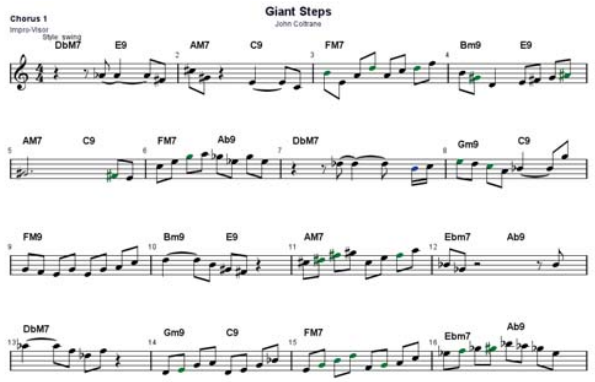


Figure 7. Generated Coltrane-style solo for "Giant Steps"

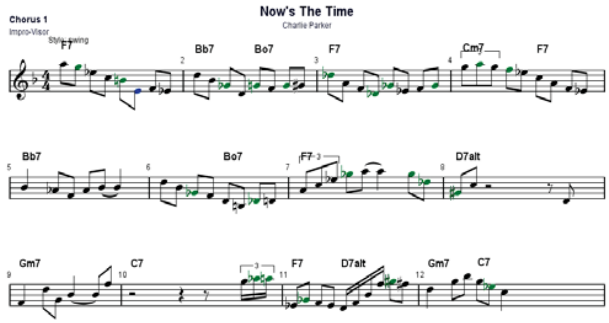


Figure 8. Generated Parker-style solo for "Now's the Time"

8. QUALITATIVE RESULTS

For short solos, we found that our algorithm's compositions usually sound like a capable jazz soloist and occasionally like a convincing imitation of an artist. For short solos of 4 to 8 bars, results were regularly very good and could be easily mistaken for the original artist, but longer solos tended to lack a sense of direction. Four-gram Markov chains produced longer coherent passages than bigram and trigram models and were able to generate very good 12 or 16 measure solos about 25 percent of the time. Given our lack of a large data set (our largest set was about 400 measures of Charlie Parker solos), higher order n-grams gave no more information than the 4-gram model. We hypothesize that with a much larger dataset, higher n-gram models would yield more coherent solos of longer duration.

9. EXPERIMENTAL RESULTS

To measure our method's effectiveness at style emulation, we set up an experiment to determine whether or not test subjects could match the styles of three prominent jazz trumpet players with solos composed in the style of each player. We inferred grammars for Clifford Brown, Miles Davis, and Freddie Hubbard from 72 bars of solos from each. We then played for the subjects one clip from each artist and one clip generated from each grammar, with each computer solo generated over the same tune ("Bye Bye Blackbird"). Without revealing the names of the artists, we asked the subjects to match the artists from the computer-composed solos with the human players. We also asked subjects to qualitatively indicate how close the resemblance was by "Not close," "Somewhat close," "Quite close," or "Remarkably close."

Out of 20 test subjects, 95 percent correctly identified Clifford Brown, 90 percent identified Miles Davis, and 85 percent identified Freddie Hubbard. Of the same subjects, 85 percent correctly matched all 3 solos. All subjects characterized the resemblance to the original artists as either "Somewhat close" or "Quite close," with 9 votes to "Somewhat close," 10 to "Quite close" and 1 unable to decide. On a scale of 1 to 10, 50 percent ranked their own musical knowledge between 2 and 5, and 50 percent between 6 and 9.

10. FUTURE WORK

A more convincing test of our method would be an experiment to determine whether listeners, particularly jazz musicians, can tell the difference between a solo generated by a learned grammar and a human composed solo. Improvements to both our musical representation and our algorithm could be made to achieve good results in such a test.

In terms of musical representation, we currently determine appropriate scales (and note categories) only by one chord. Some chords fit into several keys though, so we could make better scale choices by looking at adjacent chords. Also, examining more specific information about notes in conjunction with the note categories, such as interval from the root of a chord, could prove to be beneficial.

The greatest weakness of our generated long solos is their lack of global structure. A more conclusive evaluation of the effectiveness of n-grams for global structure could be done given a sufficiently large data set. Another approach that could be explored is to infer the high level structure of a generated solo on a particular solo from the training set. We could represent an outline for a solo with a sequence of clusters and create new solos based on the outline by choosing different cluster representatives than were in the original.

11. CONCLUSIONS

The ability of our method to generate solos that sound similar to the artist from the training data, yet distinct from any particular solo, shows that our method of data abstraction is effective. The combination of contours and note categories seems to balance similarity and novelty sufficiently well to be characterized as jazz. In addition, clustering appears to be a workable algorithm for grouping fragments of melodies. Markov chains were effective in structuring solos, however, additional global structure is desirable for providing intra-solo coherence.

12. ACKNOWLEDGMENT

This work was supported by grant 0753306 from the National Science Foundation.

13. REFERENCES

[1] Pedro P. Alcazar and Enrique Vidal-Ruiz, "Learning Regular Grammars to Model Musical Style: Comparing Different Coding Schemes", *Proc. 4th ICGI*, 211-222, 1998.

[2] Charles Ames, "The Markov Process as a Compositional Model: A Survey and Tutorial", *Leonardo*, **22** (2), 175-187, 1989.

[3] David Baker, *How To Play Bepop 1*, Alfred Publishing Co., Inc., Van Nuys, CA, 1988.

[4] Bernard Bel, "Pattern Grammars in Formal Representations of Musical Structures", *Proc. 11th Int'l Joint Conference on Artificial Intelligence, Workshop on AI & Music*, August, 1989.

[5] Chuan-Weng Chang and Hewijin Christine Jiau, "Extracting Significant Repeating Figures in Music by Using Quantized Melody Contour", *Proc. Eighth IEEE Int'l Symposium on Computers and Communication* (ISCC'03), 2003.

[6] David Cope, "Computer Modeling of Musical Intelligence in EMI". *Computer Music Journal*, **16** (2), 1992.

[7] David C. De Roure and Steven G. Blackburn, "Content-based navigation of music using melodic pitch contours", *Multimedia Systems*, **8** (3), October, 2000.

[8] Shlomo Dubnov, Gerard Assayag, Olivier Lartillot, and Gill Bejerano, "Using Machine-Learning Methods for Musical Style Modeling", *Computer*, **36** (10), 73-80, October, 2003.

[9] Douglas Eck and Jasmin Lapalme, "Learning Musical Structure Directly from Sequences of Music", *Tech. Rept. 1300, Universite de Montreal DIRO*, 2008.

[10] Red Garland, on "Miles Davis in Person, Friday Night at the Blackhawk, San Francisco", Columbia Records, 1961.

[11] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A *k*-Means Clustering Algorithm", *Applied Statistics*, **28** (1), 100–108, 1979.

[12] Impro-Visor, Jazz Improvisation Advisor, `http://www.impro-visor.com`.

[13] Yong-Kyoon Kang, Kyong-I Ku, Yoo-Sung Kim, "Extracting Theme Melodies by Using a Graphical Clustering Algorithm for Content-Based Music Information Retrieval", *Proc. 5th East European Conference on Advances in Databases and Information Systems*, 84-97, 2001.

[14] Robert M. Keller and David R. Morrison, "A Grammatical Approach to Automatic Improvisation", *Proc. Fourth Sound and Music Conference*, Lefkada, Greece, July, 2007.

[15] Youngmoo E. Kim, Wei Chai, Ricardo Garcia, and Barry Vercoe, "Analysis of a Contour-based representation for Melody", *Proc. Int'l Symposium on Music Information Retrieval*, 2000.

[16] John McCarthy, "Recursive functions of symbolic expressions and their computation by machine", *Comm. ACM*, **3** (1), 184-195, 1960.

[17] Jon McCormack, "Grammar-Based Music Composition". In Stocker et al, eds. *Complex Systems 96: from local interactions to global phenomena*, 321-336. IOS Press, 1996.

[18] François Pachet, "Surprising Harmonies", *Int'l Journal on Computing Anticipatory Systems*, 1999.

[19] George Papadopoulos and Geraint Wiggins, "AI Methods for Algorithmic Composition: A Survey, a Critical View and Future Prospects", in *AISB Symposium on Musical Creativity*, 1999.

[20] Karsten Verbeurgt, Michael Dinolfo, and Mikhail Fayer, "Extracting patterns in music for composition via Markov chains", *Proc. 17th International Conference on Innovations in Applied Artificial Intelligence*, 1123-1132, 2004.