

# Machine Learning Applied to Musical Improvisation

Robert M. Keller  
Harvey Mudd College

Constructive Machine Learning Workshop  
NIPS 2013, 10 December 2013  
Lake Tahoe, Nevada, USA

# Abstract

Development of music education software inevitably leads to questions of how to acquire musical knowledge to be made available to the student user. I will describe machine learning of patterns for accompaniment styles and grammars for improvisation, based on melodic abstraction, clustering, and chaining. I will also discuss supervised and unsupervised approaches to improvising over chord progressions using neural network. Finally, I will mention a challenging unsolved application: learning to classify idiomatic patterns in chord progressions.

# Collaborators for Machine Learning Aspects

Jon Gillick

Kevin Tang

Jim Herold

Brandy McMenemy

Sayuri Soejima

Hayden Blauzvern

Greg Bickerman

Sam Bosley

Peter Swire

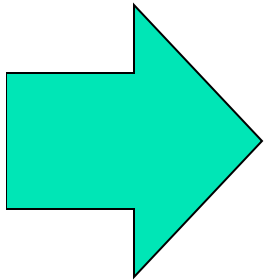
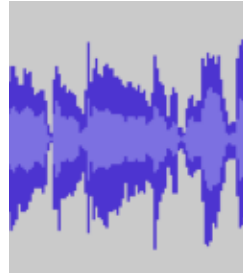
Kevin Choi

# Context: Educational Music Software

- Intelligent Music Software Project at Harvey Mudd College
  - **Impro-Visor**
    - Style pattern learning
    - Melodic grammar learning
    - Melody generation using neural network critic
  - **RBM-provisor: Using Deep-Belief Networks**
    - Melody generation using unsupervised learning

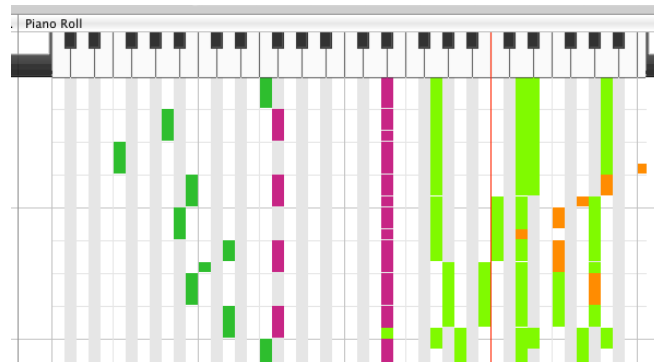
# Two Major Music Universes

- **Audio** (mp3, wav, AIFF, au, ...)



- **Symbolic**

MIDI = “Musical Instrument Digital Interface”



# Example: SmartMusic MakeMusic, Inc.

- Provides feedback for student practice sessions (“used by over 75,000 students worldwide”)
- <http://www.youtube.com/watch?v=xhYXO6TPKw4>
- Developed by Prof. Roger Dannenberg at CMU.
- Proprietary



# Learning

- Intelligent software can also “learn”, so as to *improve* its ability to make decisions beneficial to the user.
- Also ideally, humans can learn from the software, whether or not the software learns.

# Music Plus One (formerly Music++)

Prof. Chris Raphael, Indiana University

- Uses Hidden Markov Models
- Virtual orchestra anticipates player's tempo, follows retakes, etc.
- [http://www.music.informatics.indiana.edu/~craphael/music\\_plus\\_one/index.html](http://www.music.informatics.indiana.edu/~craphael/music_plus_one/index.html)





# Creativity

Ideally, intelligent music software can also “create”, i.e. use its ability to make decisions to produce new results that will inform or intrigue the user.

# GenJam (Genetic Jammer)

Prof. Al Biles, Rochester Inst. of Technology

- Improvises jazz solos
- Trades interactively with human soloist.
  - <http://www.youtube.com/watch?v=xWHU8uE043g>
- Learning based on Genetic Algorithm
- Proprietary code



# HMC Intelligent Music Software Project

- Oriented toward helping musicians learn to improvise
- Focus is on jazz education, but not limited to jazz

Impro-Visor®



# Impro-Visor

- Short for “Improvisation Advisor”.
- A software “workbook” that can learn to improvise by:
  - Helping users **write out** solos
  - **Creating** solos while **trading** with user
  - Playing backing accompaniments

# Impro-Visor Parameters

- Much musical information is in the form of user-editable text files:
  - **Vocabulary** defines
    - Scales, Chords, Cells, Idioms, Licks, Quotes
  - **Styles** govern
    - How accompaniment is played and sound
  - **Grammar** creates melodies
    - Somewhat in the style of specific players
  - **Leadsheet** specifies
    - Chord progression
    - Melody, solos

# Leadsheet vs. Sheet Music



Sheet music for a single bar. The top staff is a vocal line in treble clef with a key signature of three sharps (F#, C#, G#) and a 3/4 time signature. The lyrics are: "It's close to mid", "You hear the door\_\_", "They're out to get\_\_". Above the staff is a guitar chord diagram for F#7. The bottom staff is a piano accompaniment in grand staff (treble and bass clefs) with the same key signature and time signature, showing a melodic line in the bass clef and a sustained chord in the treble clef.

1 bar of **sheet music**



Lead sheet for a single bar. The top staff is a vocal line in treble clef with a key signature of three sharps (F#, C#, G#) and a 4/4 time signature. The lyrics are: "It's close to mid", "You hear the door\_\_", "They're out to get\_\_". Above the staff is a guitar chord diagram for F#7. The bottom staff is a piano accompaniment in grand staff (treble and bass clefs) with the same key signature and time signature, showing a melodic line in the bass clef and a sustained chord in the treble clef.

1 bar of a **leadsheet**

In a leadsheet, the accompaniment aspect is left to the performer.

# Impro-Visor Leadsheet View

The screenshot displays the Impro-Visor software interface for the song "After You've Gone" by Turner Layton. The window title is "Impro-Visor: After You've Gone". The menu bar includes File, Edit, Transpose, View, Play, Utilities, Window, Grammar: LeeMorgan, Preferences, and Help. The toolbar contains various icons for file operations, playback, and editing, along with buttons for "Generate", "Thaw", "B/W", "Simple", and "No Beam". The "Chord Font" is set to 16. The "Program Status" section includes a "Click in notes, or type in textual entry field" button. The playback controls show a count of 0, a playback location from 0:00 to 1:30, a looping button set to 2, a volume slider, a tempo of 160.0 BPM, and other settings like Transpose (0), Bars/Chorus (20), Tracker Delay (0), and Parallax (0). The "Textual Entry" field contains "FmM7" and a "Clear" button. The main area shows the leadsheet for "After You've Gone" (1918) in 4/4 time with a swing style. The melody is written on a treble clef staff with green notes. Chords are indicated above the staff: FM7, FmM7, CM7, A7, D7, G7, CM7, C7, FM7, FmM7, CM7, A7, Dm7, A7, Dm7, Bb7, CM7, E7, Am7, D7, CM7, G7, CM7, and C7. The interface also shows "Chorus 1", "Chorus 2", and "Chorus 3" tabs.



# User Constructing a Solo

The screenshot shows the Impro-Visor software interface. The title bar reads "Impro-Visor: After You've Gone". The menu bar includes File, Edit, Transpose, View, Play, Utilities, Window, Grammar: My, Preferences, and Help. The toolbar contains icons for file operations, playback, and editing, along with buttons for "Generate", "Freeze", "B/W", "Simple", and "No Beam". The "Program Status" section indicates "Click in notes, or type in textual entry field". The playback controls show a "Count" of 0:00, a "Playback Location" slider from 0:00 to 1:30, "Looping" set to 2, "Volume" control, "Tempo (Beats per Minute)" set to 160.0, "Transpose" set to 0, "Bars/Chorus" set to 0, "Tracker Delay" set to 0, and "Parallax" set to 0. A "Textual Entry" field is visible with a "Clear" button. The main score area is titled "Chorus 1" and shows a musical score in 4/4 time with a "Style: swing" setting. The score consists of three staves. The first staff shows a melody line with notes and rests, with a vertical red line indicating the current playback position. The second and third staves show chord progressions. The chords are: FM7 (measures 1-2), FmM7 (measures 3-4), CM7 (measures 5-6), A7 (measures 7-8), D7 (measure 9), G7 (measure 10), CM7 (measures 11-12), and C7 (measure 13). The score is labeled "Chorus 1" and "Style: swing".

## Entry Options:

- Point-and-click
- Cut-and-paste
- Textual
- Midi-keyboard



# Note Coloration: Chord-Related

Part of the “Advisor” aspect of Impro-Visor

**Blue:** Half-step away from chord or color (called “approach” tone).

**FmM7** **CM7**

A musical staff in bass clef showing a sequence of notes. The first two notes are black. The next two are green. The fifth note is blue with a sharp sign. The sixth note is red with a flat sign. A bar line is after the sixth note. A '3' is written above the staff. A blue arrow points from the text box above to the blue note. A red arrow points from the text box below to the red note. A black arrow points from the text box below to the first black note. A green arrow points from the text box below to the first green note.

**Red:** None of the others (“outside”).

**Green:** tone not in the chord, but sonorous with it (called “color” tone).

**Black:** tone in the chord

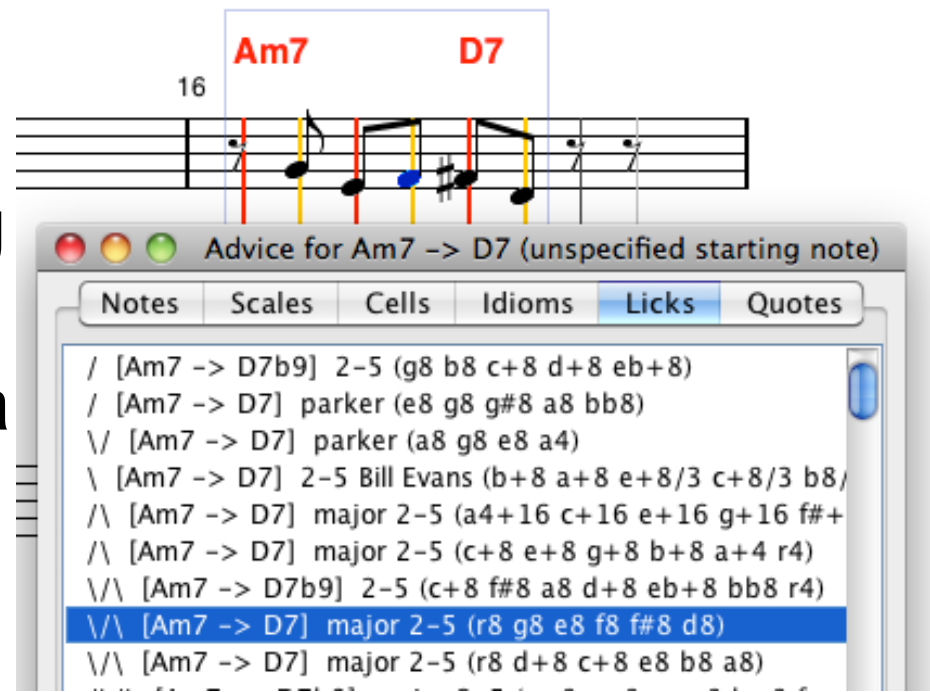
# A Helpful Insight

- Note coloration **categories**, originally intended to educate users, can be useful for **machine learning** aspects.
- Use in Grammar Learning
- Use in Critic Development

# Generating Licks for Examples

- **Lick** = a short melodic phrase
  - sometimes idiomatic
  - sometimes original

- Prior to introducing lick generation, Impro-Visor used a **database** to store lick suggestions.

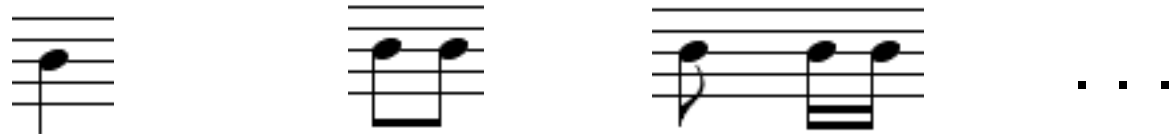


The image shows a musical staff with a lick starting at measure 16. The lick consists of the following notes: G4 (quarter), A4 (quarter), B4 (quarter), C5 (quarter), B4 (quarter), A4 (quarter), G4 (quarter), F#4 (quarter), E4 (quarter), D4 (quarter). Above the staff, the chords Am7 and D7 are indicated. Below the staff is a software window titled "Advice for Am7 -> D7 (unspecified starting note)". The window has tabs for Notes, Scales, Cells, Idioms, Licks, and Quotes. The Licks tab is selected, showing a list of suggestions. The selected suggestion is: \\ [Am7 -> D7] major 2-5 (r8 g8 e8 f8 f#8 d8).

```
Advice for Am7 -> D7 (unspecified starting note)
Notes Scales Cells Idioms Licks Quotes
/ [Am7 -> D7b9] 2-5 (g8 b8 c+8 d+8 eb+8)
/ [Am7 -> D7] parker (e8 g8 g#8 a8 bb8)
\ [Am7 -> D7] parker (a8 g8 e8 a4)
\ [Am7 -> D7] 2-5 Bill Evans (b+8 a+8 e+8/3 c+8/3 b8/
/\ [Am7 -> D7] major 2-5 (a4+16 c+16 e+16 g+16 f#+
/\ [Am7 -> D7] major 2-5 (c+8 e+8 g+8 b+8 a+4 r4)
\\ [Am7 -> D7b9] 2-5 (c+8 f#8 a8 d+8 eb+8 bb8 r4)
\\ [Am7 -> D7] major 2-5 (r8 g8 e8 f8 f#8 d8)
\\ [Am7 -> D7] major 2-5 (r8 d+8 c+8 e8 b8 a8)
```

# Probabilistic Grammar Illustration

- We could fill a beat with a variety of rhythms:



- Let B denote one beat of music
- A grammar represents all of these possibilities:

$B \rightarrow X4$                       4 means quarter note  
 $B \rightarrow X8 X8$                       8 means eighth note  
 $B \rightarrow X8 X16 X16$                       etc.

Here X4, X8, X16 are understood “terminal” symbols, while B is a non-terminal to be expanded.

# Probabilistic Grammar Illustration

- Assign a probability to the various choices
- Probabilities will then dictate a prevalent style



- A grammar represents a distribution of these possibilities:

$B \rightarrow X4$	$p = 0.3$	common
$B \rightarrow X8 X8$	$p = 0.6$	frequent
$B \rightarrow X8 X16 X16$	$p = 0.1$	rare

# Grammars Can Exhibit Hierarchy and Recurrence


- Instead of

$B \rightarrow X4$	$p = 0.3$	common
$B \rightarrow X8 X8$	$p = 0.6$	frequent
$B \rightarrow X8 X16 X16$	$p = 0.1$	rare

- Use

$B \rightarrow X4$	$p = 0.3$	common
$B \rightarrow C C$	$p = 0.7$	frequent
$C \rightarrow X8$	$p = 0.8$	very frequent
$C \rightarrow X16 X16$	$p = 0.2$	rare

- Generates

				
$p = 0.3$	$p = 0.448$	$p = 0.112$	$p = 0.112$	$p = 0.028$

# Recurrence Allows a Grammar to Fill an Arbitrary Number of Beats

- $R \rightarrow B R$                       One beat, then more
- $R \rightarrow \text{empty}$                       No more
  
- So  $R$  can produce  $B$ ,  $BB$ ,  $BBB$ ,  $BBBB$ ,  
etc.

# Abstract vs. Real Melodies

A real melody  
on the staff  
and as text

**Bb13**      **Bo7**

c+8 ab8 bb8 e8 db4 d8 f8

C = Chord tone  
L = coLor tone  
8 = 8th note  
4 = quarter noter

Abstract melody as text: C8 C8 C8 C8 L4 C8 C8

Two other melodies with  
the same abstract melody

d+8 bb8 f8 ab8 g4 b8 f+8

**Bb13**      **Bo7**

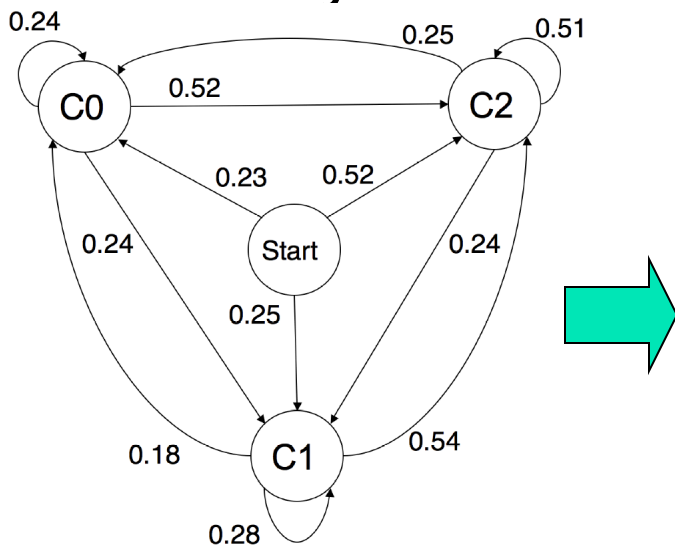
g8 bb8 f8 ab8 db+4 f+8 g#+8

**Bb13**      **Bo7**

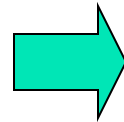


# Markov Chains in Grammars

- Recurrent productions allow us to embed an arbitrary Markov chain in the grammar.
- The reason for wanting this will be explained shortly.



Markov chain



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

Grammar

# A Complete Grammar (Terminals in Bold)

(startsymbol P)  
(base (P 0) () 1.0)  
(rule (M4) (**A4**) 0.01)  
(rule (M4) (**C4**) 0.2)  
(rule (M4) (**C4**) 0.1)  
(rule (M8) (**A8**) 0.01)  
(rule (M8) (**C8**) 0.4)  
(rule (M8) (**C8**) 0.2)  
(rule (M8) (**C8**) 0.1)  
(rule (N2) (**C2**) 1.0)  
(rule (N4) (**M4**) 0.75)  
(rule (N4) (**R4**) 0.25)  
(rule (N8) (**M8**) 0.9)  
(rule (N8) (**R8**) 0.1)  
  
(rule (P Y) (Seg1 (P (- Y 120))) 0.0010)  
(rule (P Y) (Seg2 (P (- Y 240))) 0.25)  
(rule (P Y) (Seg4 (P (- Y 480))) 0.75)  
(rule (Seg1) (C4) 1.0)  
(rule (Seg2) (N2) 0.06)  
(rule (Seg2) (N8 C4.) 0.3)

(rule (Seg2) (V2) 0.3)  
(rule (Seg2) (V4 V4) 0.6)  
(rule (Seg2) (V8 N4 V8) 0.12)  
(rule (Seg2) (V8 V8 V8 V8) 0.6)  
(rule (Seg4) (C4. N8 Seg2) 0.1)  
(rule (Seg4) (C4/3 C4/3 C4/3 Seg2) 0.02)  
(rule (Seg4) (Seg2 C4/3 C4/3 C4/3) 0.02)  
(rule (Seg4) (Seg2 V4 V4) 0.52)  
(rule (Seg4) (V8 N4 N4 N4 V8) 0.01)  
  
(rule (V2) (**C16 C16 C16 C16** M4) 0.05)  
(rule (V2) (**C16/5 C16/5 C16/5 C16/5 C16/5** M4) 0.0050)  
(rule (V2) (**C8 C8 C8 C8**) 0.3)  
(rule (V2) (**C8/5 C8/5 C8/5 C8/5 C8/5**) 5.0E-4)  
(rule (V4) (**C8/3 C8/3 A8/3**) 0.01)  
(rule (V4) (**C8/3 C8/3 C8/3**) 0.05)  
(rule (V4) (**C8/3 C8/3 C8/3**) 0.02)  
(rule (V4) (N4) 0.22)  
(rule (V4) (V8 V8) 0.72)  
(rule (V8) (**C16 A16**) 0.01)  
(rule (V8) (N8) 0.99)

# Grammar Learning Feature

- Impro-Visor can **learn a grammar** by examining one or more transcribed solos.
- For greater coherence special construct called a ***slope*** is introduced, from which **melodic contours** can be constructed.
- Slopes can appear in the rules in the place of terminals.

# Slopes Encode Contours

(a) Original melody:



(b) Melody contour:



(c) Abstract melody using slopes ( $\Delta$ 's):

(R8 C8 ( $\Delta$  -9 -9 A16) ( $\Delta$ 1 3 C16 C16 C16 C8) ( $\Delta$ -12 -12 C8) ( $\Delta$ 1 4 C8 A8)  
 ( $\Delta$  -4 -1 L8 C8 C8 A8 C8) ( $\Delta$ 12 12 C8) ( $\Delta$ -12 -2 C8 C8))

# Grammar Learning Algorithm

enables grammar to be learned from transcriptions

Transcription of Dave Liebman's Solo on Picadilly Lilly:

**Picadilly Lilly**  
Dave Liebman

Chorus 1  
Dave Liebman: swing

Chords: F#M7, Bm7, Dm7, C#7b9, F#M7, G7b9, F#M7, Bm7

Notes: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

Transcription in Text

Aquamacs - PicadillyLillyLiebman.ls

```
f+8 d#+8 r2+4+8
f+8 d#+8 c#+8 f+8 f+8 d#+8 c#+8
f#+8 d+8 c#+8 a8 b8 e+8 r8 c#+8
d+8 e+8 f#+8 a+8 c#++8 e++4 d++4
c++2 a+16 g+16 e+8 f+4+8
r2+4+8
c#+8 a#8 c#+8 c+8 a8 bb8 c+8
b8 g8 e8 g8 eb4+8 f16 e16
f4 g#8 f8 a#4 c+8 a8
d#+4 f+8 d#+8 g#+4 a#+8 g#+8
c++16 c#++16 e++8 b+8 c#++8 a+8 a+8 g#+8 c#+8
f#+8 g#+8 a+8 b+8 d#+16 e+4+16 d+8
f+8 a+8 e+8 d+4+8 r1+2+4+8
c#++8
a#+8 g#+8
e+16 f+8+16 g#+8 d#+8 r8 d#+8 c#+8 a8
b8 d+8 f#+8 a+8 g#+2
r4+8 e+8 f#+4 a+1
```

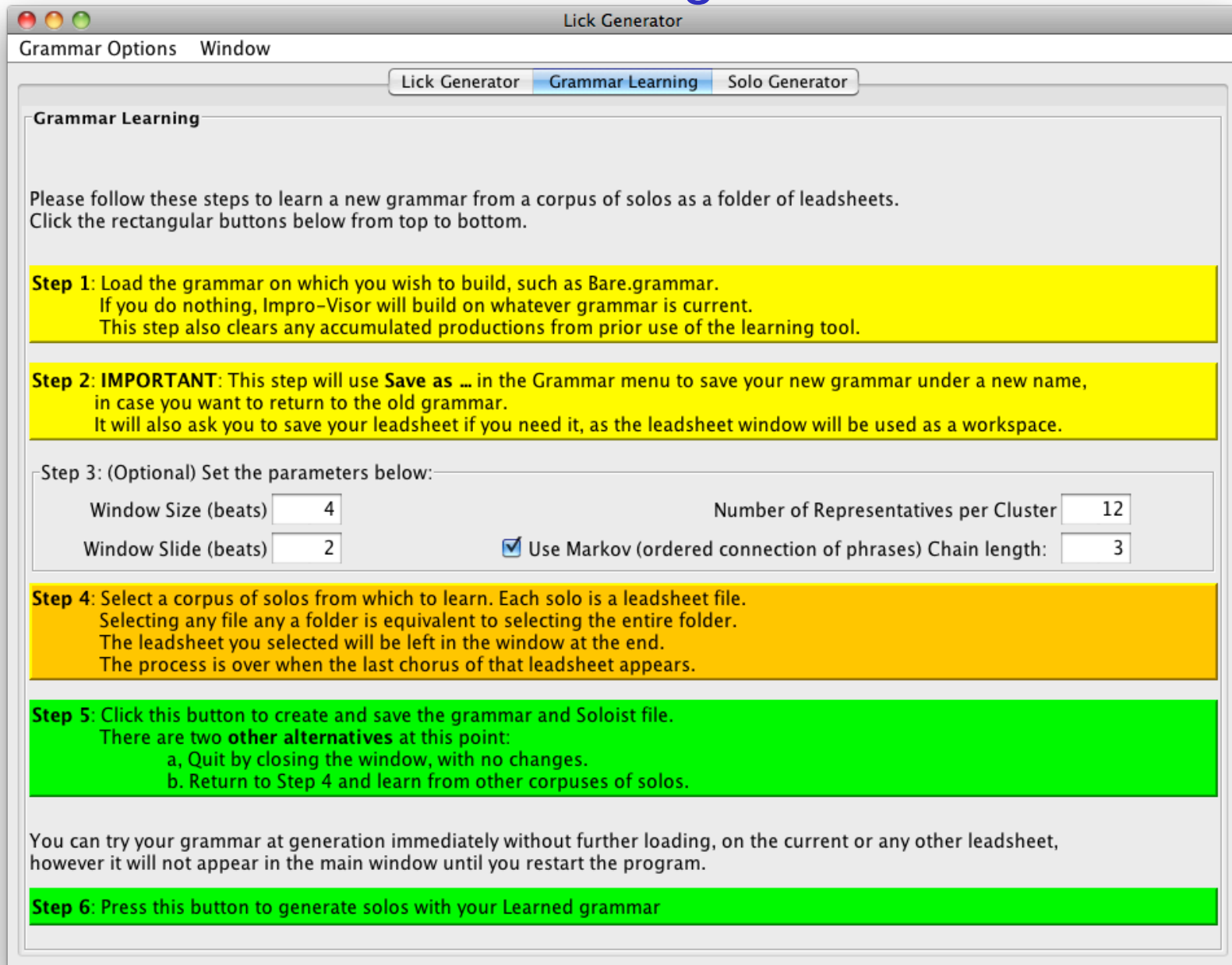
-.\*\*\* PicadillyLillyLiebman.ls 13% (63,1) (Text)

Learned Grammars

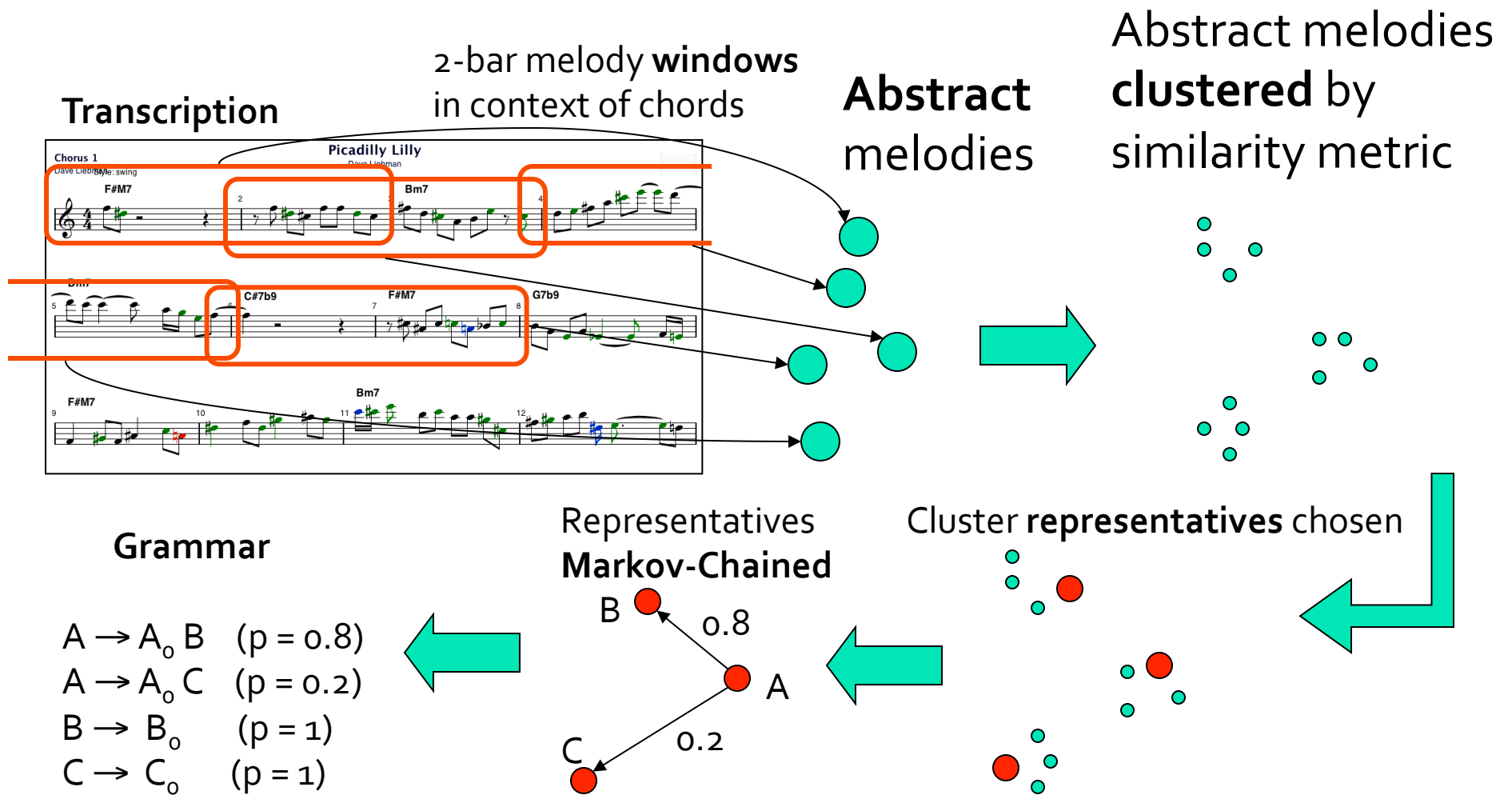
My grammar Preferences H

- Backup
- Bare
- BillEvans
- CharlieParker
- CliffordBrown
- ColemanHawkins-Ballads
- DaveLiebman**
- DexterGordon
- DizzyGillespie
- FreddieHubbard
- JimmyHeath
- JJJJohnson
- JohnColtrane
- KeithJarrrett
- LeeMorgan
- MilesDavis
- My
- MyFours
- Outside
- RedGarland
- TomHarrell-Waltzes
- TomHarrell
- Zoo

# Grammar Learning Interface



# From Transcription to Grammar



# Example: Dave Liebman Grammar Excerpt (The full grammar is over 1000 lines)

```
(startsymbol P)
(base (P 0) () 1.0)
(rule (P Y) (Seg1 (P (- Y 120))) 1.0)
(rule (Seg1) (C4) 1.0)
(rule (P Y) ((START 1) (P (- Y 480))) 1.0)
(rule (P Y) ((START 2) (P (- Y 960))) 10.0)
(rule (P Y) ((START 4) (P (- Y 1920))) 100.0)
(rule (P Y) ((START 8) (P (- Y 3840))) 1000.0)

(rule (START Z) ((Cluster0 Z)) 0.03)
(rule (START Z) ((Cluster1 Z)) 0.02)
(rule (START Z) ((Cluster2 Z)) 0.07)
(rule (START Z) ((Cluster3 Z)) 0.08)
...
(rule (START Z) ((Cluster28 Z)) 0.08)

(base (Cluster0 0) () 1)
(base (Cluster1 0) () 1)
...
(base (Cluster28 0) () 1)
```

```
(base (Cluster0to3 0) () 1)
(base (Cluster0to4 0) () 1)
(base (Cluster0to11 0) () 1)
...
(base (Cluster28to28 0) () 1)

(rule (Cluster0 Z) (Q0 (Cluster0to3 (- Z 1))) 0.33)
(rule (Cluster0 Z) (Q0 (Cluster0to4 (- Z 1))) 0.11)
...
(rule (Cluster28to28 Z) (Q28 (Cluster28to13 (- Z 1))) 0.33)

(rule (Q0)((slope 0 0 C2)(slope -4 -4 R4+8 L8)) 0.20)
(rule (Q0)((slope 0 0 C4)(slope -2 -2 R4+8 L8)(slope 1 5 X8 A8)) 0.20)
(rule (Q0)((slope -2 -1 L2)(slope -4 -4 R4+8 L8)) 0.20)
...
(rule (Q28)((slope 2 4 L4+8)(slope -2 -1 A8 C8 L8 C8)(slope 2 2 L8)) 0.20)
```



# Style Learning in Impro-Visor

- Style Patterns are used (along with chord sequence) in creating **accompaniment**.
- Patterns are like a non-recursive grammar.
- Impro-Visor can **learn** a style specification (in its own language), given a MIDI file of a **performance** in that style.
- As with grammar learning, **clustering** is used.

# Style Patterns Represented Graphically

## Style Spreadsheet

Style Editor: african3.sty

File Edit Extract Help Window

Comments Saved with Style

Bass Attributes: High: c, Octaves: --, Nominal: c, Low: c

Chord Attributes: High: c, Octaves: --, Voicing Type: open

Melody Swing: 0.5

Chord played over pattern: Major

Comp Swing: 0.5

Play pattern when cell clicked: Mute Volume 170 BPM

File Column Edit Row Edit Cell Edit Time Play Saved Style Editor Status

Clipboard

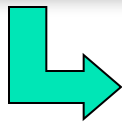
Mirrored patterns, most recent pattern at the bottom

Row Column

Double-click table cell to edit pattern. Control-click a column of table to play percussion simultaneously. Shift-click a column to use piano roll editor.

	Use	Volume	1	2	3	4	5	6	7	8	9	10
Bass Beats			3.0	2.0	0	0	0	0	0	0	0	0
Bass Weight			43	15	10	10	10	10	10	10	10	10
Bass	<input checked="" type="checkbox"/>	100	B8+32 B1	B4/3 R16								
Chord Beats			3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0
Chord Weight			38.0	3.0	2.0	15.0	10	10	10	10	10	10
Chord Push												
Chord	<input checked="" type="checkbox"/>	100	X8+32 X2	X2+4	X2+4/3	X8+32 X4						
Drum Beats			3.0	3.0	3.0	3.0	3.0	3.0	3.0	0	0	0
Drum Weight			13.0	21.0	36.0	1.0	6.0	4.0	15.0	10	10	10
Cabasa	<input checked="" type="checkbox"/>	100	X8/3 X4/2	X4/3 X4/2	X8/3 X4/2	X8/3 X4/2	X8/3 X4/2	X4/3 X4/2	X8/3 X4/2			
Crash Cymbal 1	<input checked="" type="checkbox"/>	100	R2+4	R2+4	R2+4	R4/3 X8/2	R2+4	X2+4	R2+4			
Tambourine	<input checked="" type="checkbox"/>	100	X4 X4 X4	X4 X4 X4	X4 X4 X4	X8/3 X8/2	X4 X8/3 X	R2+4	X4 X4 X4			
High Timbale	<input checked="" type="checkbox"/>	100	R2+4	R2+4	R2+4	R2+4	R4 X8/3 X	R2+4	R2+4			
Maracas	<input checked="" type="checkbox"/>	100	X8/3 X8/2	X8/3 X8/2	X8/3 X8/2	X8/3 X8/2	X8/3 X8/2	X8/3 X8/2	X8/3 X8/2			
Open Hi Conga	<input checked="" type="checkbox"/>	100	R8/3 X4/2	X8/3 X8/2	R8/3 X8/2	R8/3 X4/2	X8/3 X8/2	R2/3+8/3				
Cowbell	<input checked="" type="checkbox"/>	100	R2+4	R2+4	R2+4	X8/3 X2+	R2+4	R2+4	R2+4			
Mute Hi Conga	<input checked="" type="checkbox"/>	100	X4/3 X4/2	R4 X4/3 X	R2 X4/3 X	X4/3 X2/2	X4/3 X2/2	R4 X4/3 X	X4/3 X4/2			
Low Floor Tom	<input checked="" type="checkbox"/>	100	X2+4	R2+4	R4/3 X2+	X2+4	R4/3 X2+	R2+4	R2+4			
Low Tom	<input checked="" type="checkbox"/>	100	R2+4	R2+4/3 X	R2+4	R2+4	R2+8/3 X	R2+4/3 X	R2/3+8/3			
Hi-Mid Tom	<input checked="" type="checkbox"/>	100	R2 X4	R4 X2	R2 X4	R2 X4	R4 X8/3 X	X4/3 X4/2	X2+4			
Acoustic Snare	<input checked="" type="checkbox"/>	100	R2+4	R2+4	R2+4	R4/3 X2+	R4 X8/3 X	R2+4	R2+4			
Open Hi-Hat	<input checked="" type="checkbox"/>	100	R2+4	R2+4	R2+4	R4 X2	R2+4	R2+4	R2+4			
Closed Hi-Hat	<input checked="" type="checkbox"/>	100	X8/3 X8/2	X8/3 X8/2	X8/3 X8/2	R2/3 X8/2	X8/3 X8/2	X8/3 X8/2	X8/3 X8/2			
Acoustic Bass...	<input checked="" type="checkbox"/>	100	X4 X4 X4	X4 X4 X4	X4 X4 X4	X4 X4 X4	X4 X4 X4	X4 X4 X4	X4 X4 X4			

Cancel Save Style



## “Piano roll” for one column of spreadsheet

Piano-Roll Pattern Editor: Column 2 of african3.sty

Open Bass Bar Editor

Long vertical lines are beats. Bass, Chord, and Percussion sections are independent, not linked together.

Bass

Chord

Acoustic Bass Drum

Closed Hi-Hat

Open Hi-Hat

Acoustic Snare

Hi-Mid Tom

Low Tom

Low Floor Tom

Mute Hi Conga

Cowbell

Open Hi Conga

Maracas

High Timbale

Tambourine

Crash Cymbal 1

Cabasa

Percussion

Percussion

Percussion

Inter-Loop Delay

Pattern Last Played:

Loop Percussion

From/To Style Editor

Play Saved Pattern

Bass

Chord

Percussion

From Style Editor Column 2

To Style Editor Column 1

Resolutions

Tempo (Beats per Minute) 130

Visual (30-120 pixels per beat) 120

Time (1-120 tick marks per beat) 8

# A Different Approach to Learning

## **RBM-provisor**

- Applies **Restricted Boltzmann Machines** (RBMs) stacked as ***Deep Belief Networks*** (Geoffrey Hinton).
- RBMs are neural networks based on **probabilities** of switching, determined by unsupervised learning of synaptic weights.
- An RBM tries to learn a set of **concepts** based on a set of input samples: melodies over chords.
- RBM's stabilize to **probability distributions** reflecting those concepts, and can generate music probabilistically, as can grammars.

# Deep Belief Network Rationale

- Try to learn with **as little wired-in** musical knowledge as possible.
- Use probabilistic behavior of network to generate novelty.

# Deep Belief Networks

Geoffrey Hinton, U. of Toronto

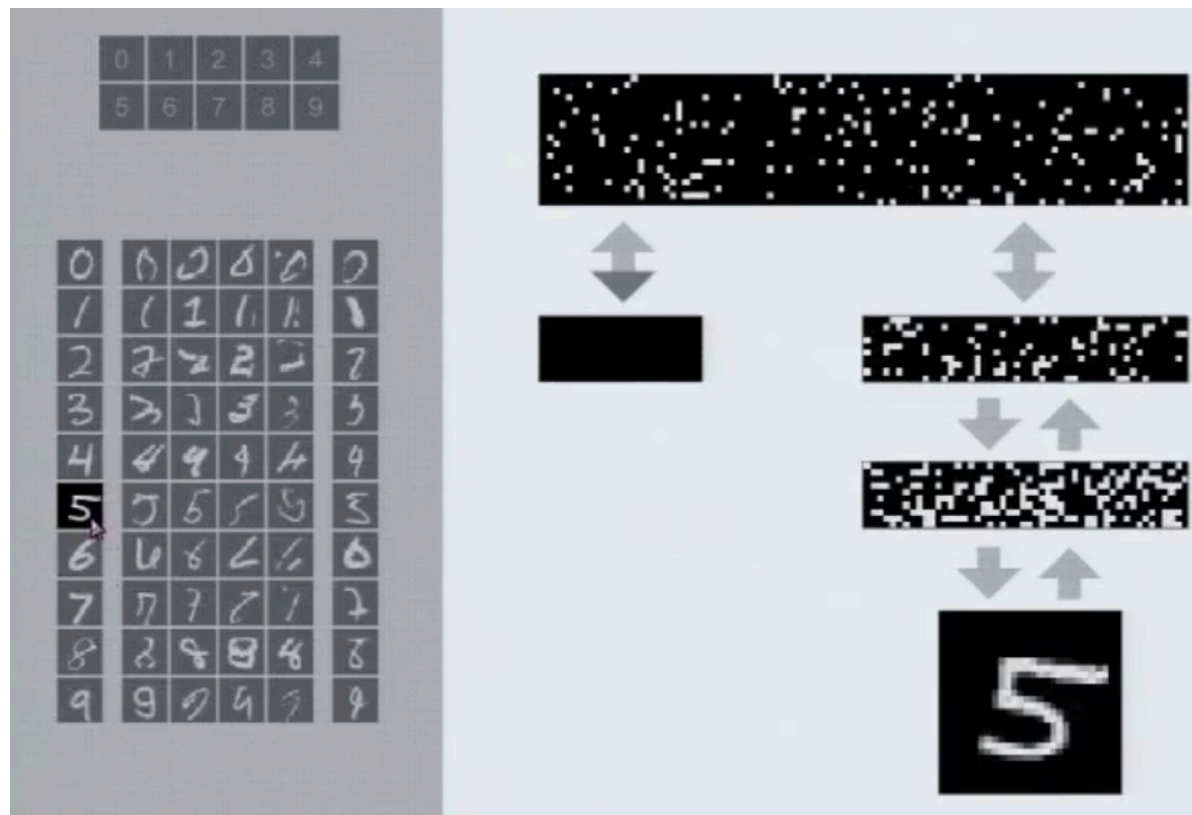
- Hinton demonstrated how a stack of RBM's can learn higher order concepts sufficient to perform tasks such as digit recognition.
- We applied a similar idea to learning concepts that **produce melodies** over chord progressions.

The Next Generation  
Of Neural Networks

Geoffrey Hinton  
November 29, 2007

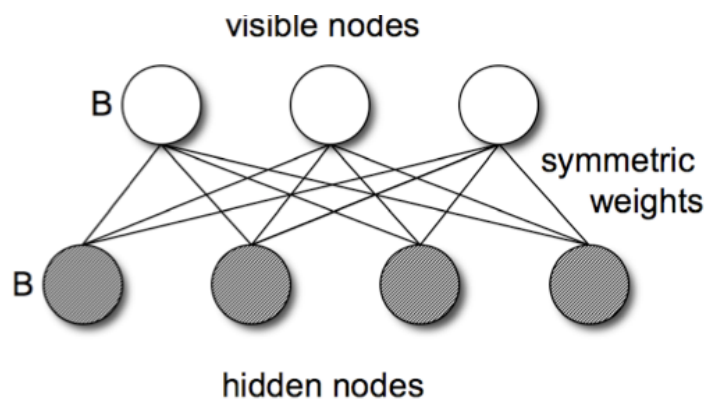
Google™

# Pattern learning and generation inspired by G. Hinton

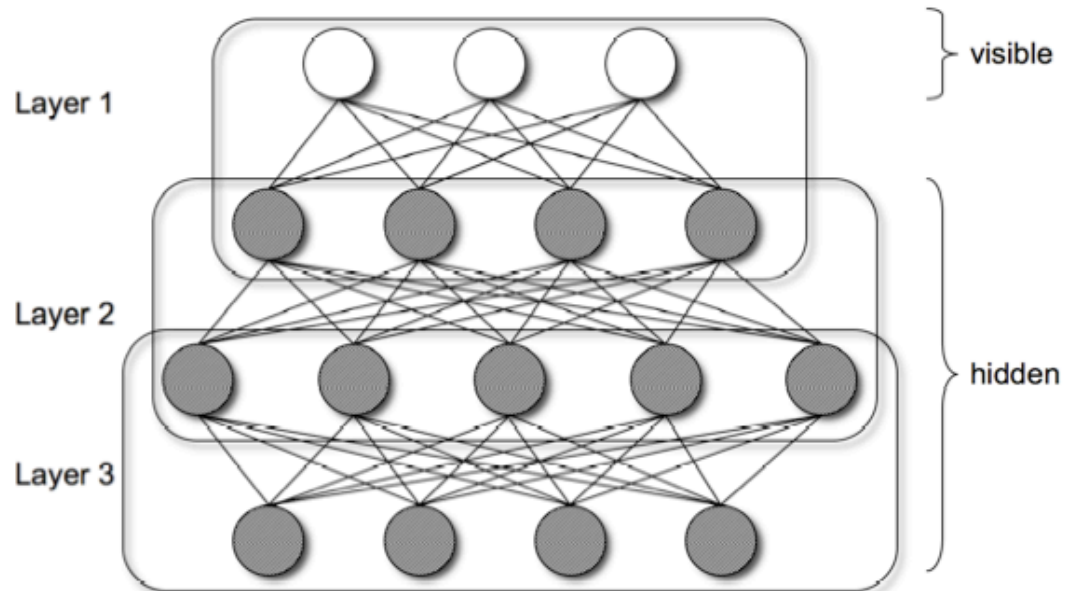


# Restricted Boltzmann Machines vs. Deep Belief Networks

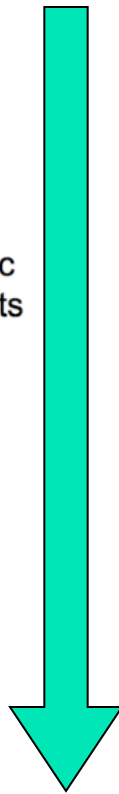
## RBM



## DBN (3-layer)



Deeper  
Concepts

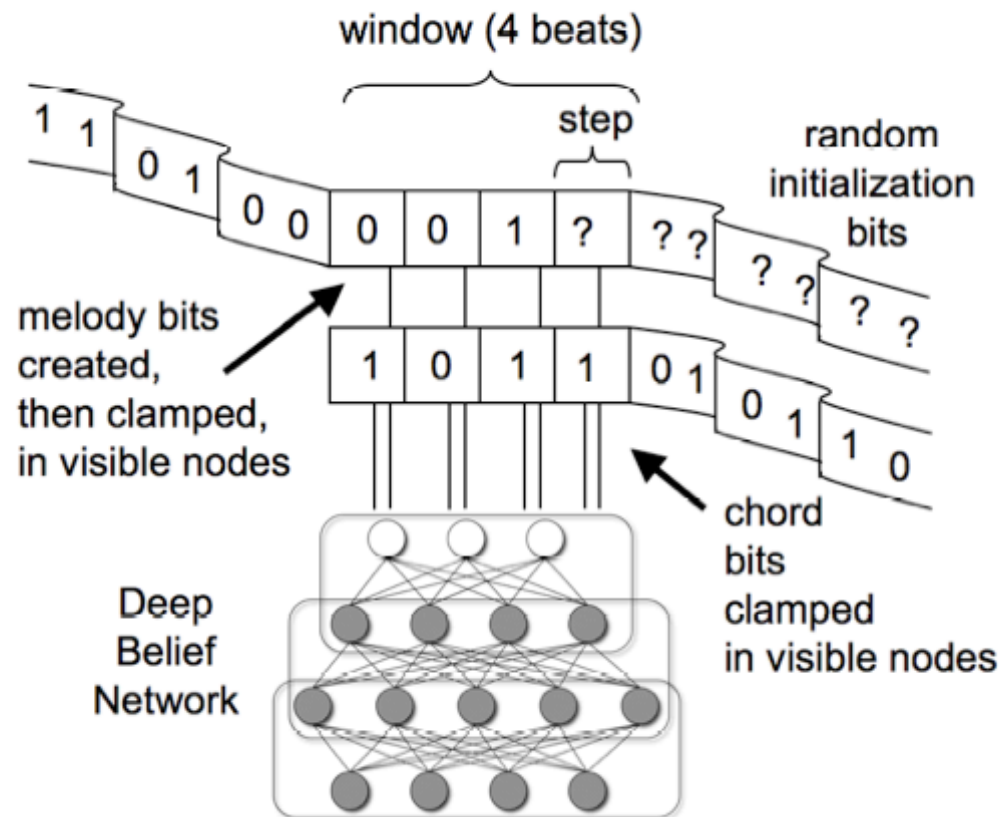






# Improvising Jazz with a Deep Belief Network

Melody Out ← Chord Progression In



# Moving Window Training Approach

Musical notation for the first line of the exercise. It consists of a single staff in 4/4 time with a treble clef. The key signature has one flat (B-flat). The notation includes a Dm7 chord at the start, followed by a melodic line with a slur and a '2' above it. This is followed by a G7 chord, a melodic line with a slur and a '3' above it, and then a C7 chord with a triplet of eighth notes. The line ends with a melodic phrase.

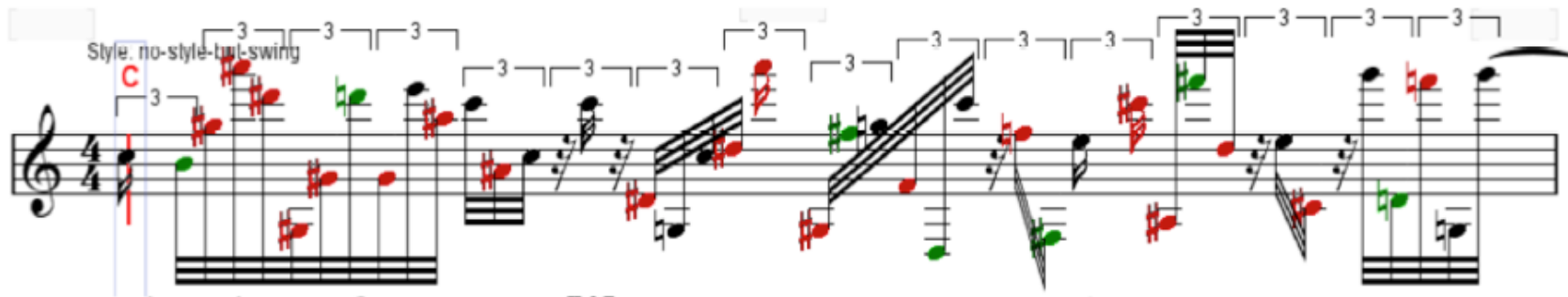
Musical notation for the second line of the exercise, identical to the first. A black rectangular box highlights the first two measures, which contain the Dm7 chord and the first melodic phrase.

Musical notation for the third line of the exercise, identical to the first. A black rectangular box highlights the second and third measures, which contain the G7 chord and the second melodic phrase.

Musical notation for the fourth line of the exercise, identical to the first. A black rectangular box highlights the third and fourth measures, which contain the C7 chord and the triplet of eighth notes.

# RBM-provisor Examples

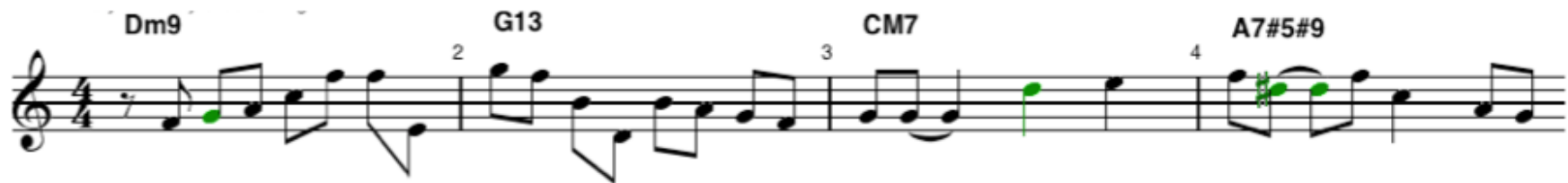
## Random output from Untrained Network



## Example from Training Set



## Output from Trained Network



# Issues with Deep Belief Approach

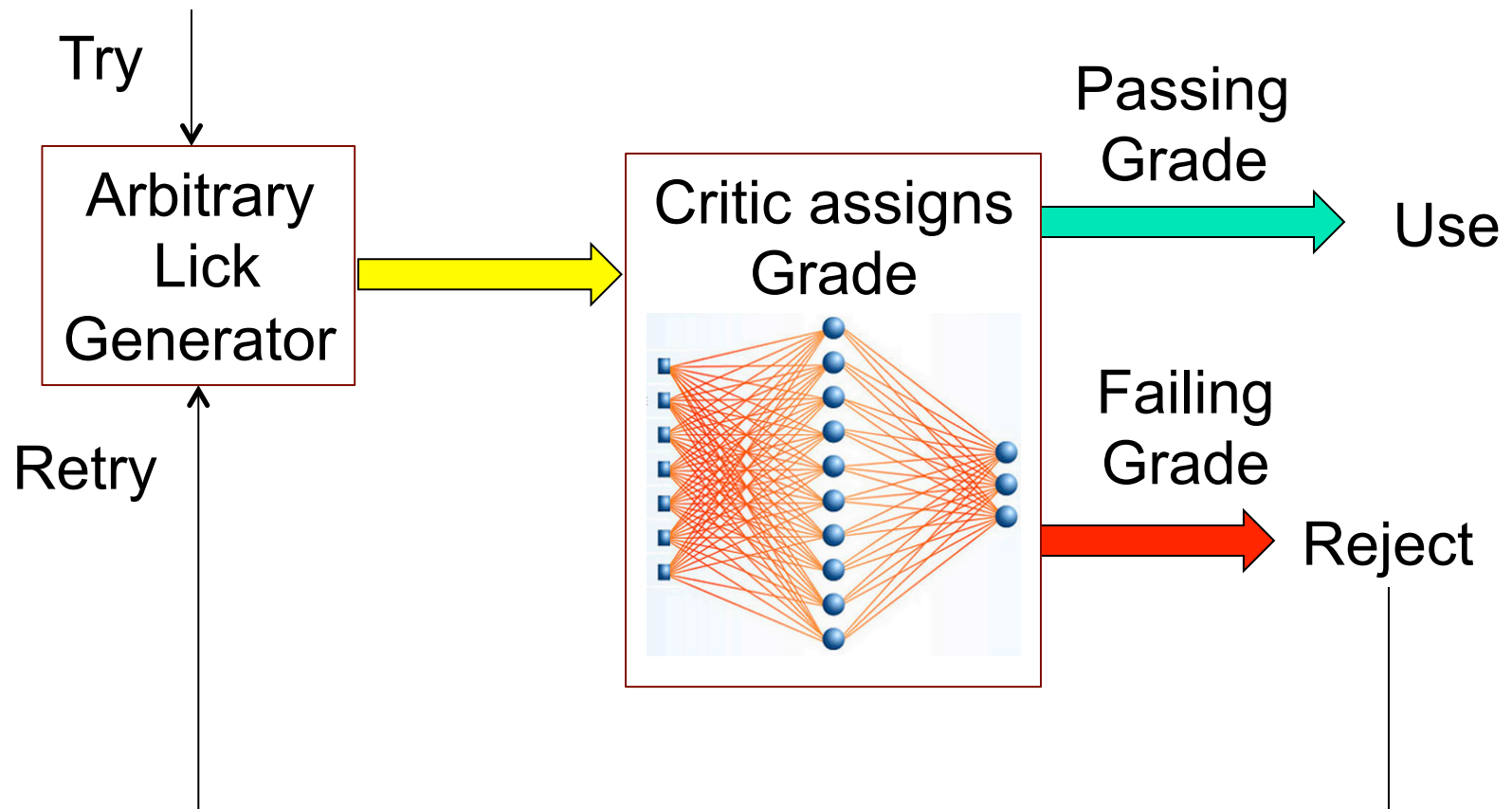
- Learning is very slow
- So far, not enough variety compared to what grammars can do.
- Thus too soon to integrate with Impro-Visor

# Neural Network Critic for Improved Improvisation

Recent work by Hayden Blauzvern and  
the speaker

Integrated into the development version of  
Impro-Visor (v 6.0, not yet released)

# Using a Critic

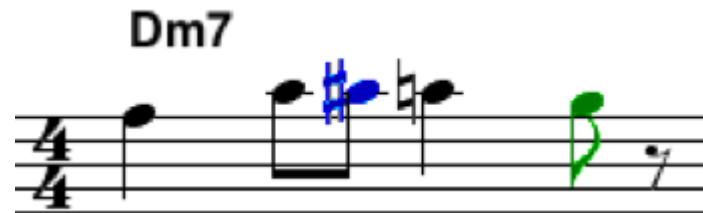


# New Critic Data Representation

(compared to RBM-provisor)

- Learn **abstract note categories** over chords rather than absolute pitches.
- Learn **pitch intervals** between successive notes rather than absolute pitches.
- Use **non-uniform sub-division** representation for note durations rather than uniform spacing and tie bits.  
(Partly motivated by accenting concerns.)

# Critic Melody Note Representation

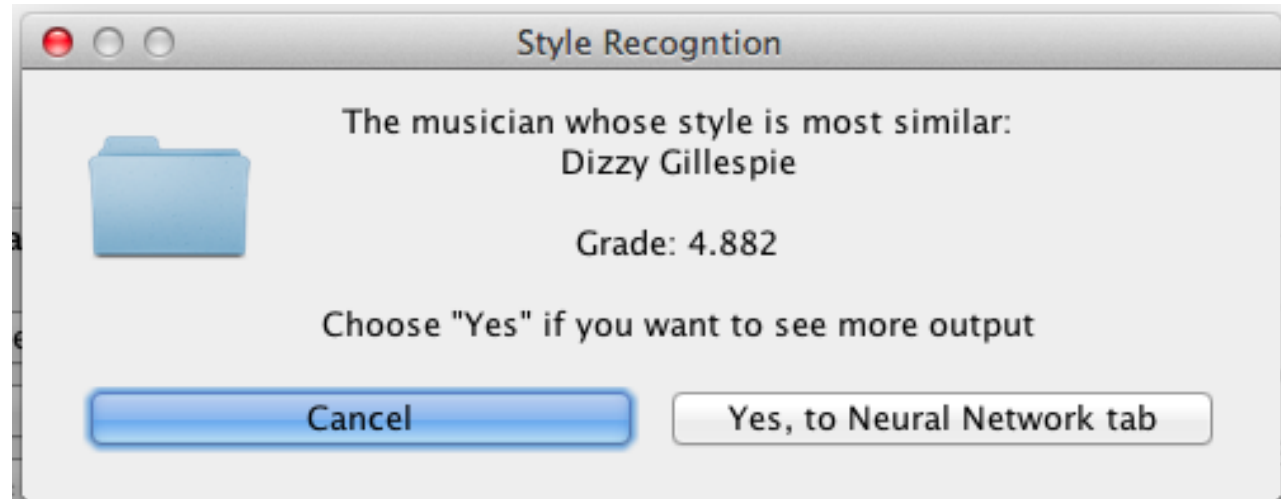


Note	Rest	Tonality	Distance	Placement	Duration
	0	1 1	0 0 0 0	1 1 1 1 1 1	0 0 1 0 0 0
	0	1 1	0 1 0 0	1 1 1 1 0 0	0 0 0 1 0 0
	0	0 1	0 0 0 1	1 1 1 0 0 0	0 0 0 1 0 0
	0	1 1	0 0 0 1	1 1 1 1 1 0	0 0 1 0 0 0
	0	1 0	0 0 1 0	1 1 1 1 0 0	0 0 0 1 0 0
	1	0 0	0 0 0 0	1 1 1 0 0 0	0 0 0 1 0 0



# Style Recognition, One NN per Style

- Grammar generated solos
  - 32 measures, 16-bar blues
- 50 generations for 22 musicians
- Grade = Confidence score



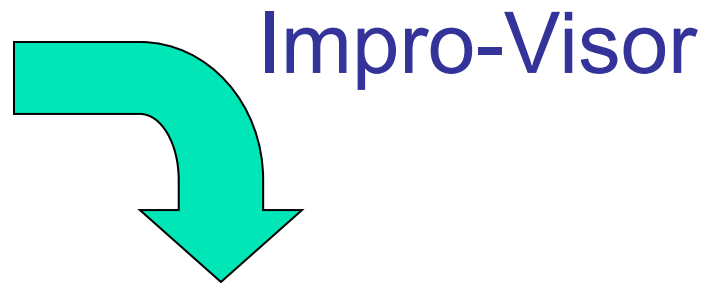
# Other Recent Work

- Automate analysis of **idiomatic harmonic sequences** (“chord bricks”) and key centers.
- Helps musicians understand tune construction.
- Helps players recognize the importance of key centers in improvisation.

# Analyzing a Tune using Bricks

## Input Chord Progression

```
Cm69 | / | Fm7 | / |
Dm7b5 | G7alt | Cm69 | / |
Ebm7 | Ab7 | DbM7 | / |
Dm7b5 | G7alt | Cm69 | Dm7b5 G7alt |
```



## Output Roadmap

Join Names  
(yellow tags)

**Open ML Problem:**

**How to learn the  
Brick Dictionary?**

Input  
Chords

Inferred  
Key

Brick  
Name

### Blue Bossa

C Minor				
On Off Minor IV				
Cm69		Fm7		Backslider
C Minor				
Sad Cadence				
Dm7b5	G7alt	Cm69	Cherokee	
Db Major				
Straight Cadence				
Ebm7	Ab7	DbM7	Downwinder	
C Minor				
Sad Cadence + ...		Minor POT		
Dm7b5	G7alt	Cm69	Dm7b5	G7alt

# Other Future Work: Bricks as a Basis for Grammar Learning

See: <http://www.cs.hmc.edu/~keller/jazz/improvisor/licks/>

Name	Chord Progression			
<a href="#">Cadence + Dropback</a>	IIIm <sup>7</sup>	V <sup>7</sup>	I	VI <sup>7</sup>
<a href="#">Sad Cadence + Dropback</a>	IIIm <sup>7</sup> b <sup>5</sup>	V <sup>7</sup>	Im	VIIm <sup>7</sup> b <sup>5</sup>
<a href="#">POT (Plain Old Turnaround)</a>	I	VI <sup>7</sup>	IIIm <sup>7</sup>	V <sup>7</sup>
<a href="#">Minor POT (Minor Plain Old Turnaround)</a>	Im	VI <sup>7</sup>	IIIm <sup>7</sup> b <sup>5</sup>	V <sup>7</sup>
<a href="#">Pullback</a>	IIIm <sup>7</sup>	V <sup>7</sup>	IIIIm <sup>7</sup>	VI <sup>7</sup>
<a href="#">Ladybird Turnaround</a>	I	bIII <sup>7</sup>	bVI	bII <sup>7</sup>

## Ladybird Turnaround Licks

Bob Keller

Style: swing

CM9 Eb13 AbM9 Db9#11

# Concluding

- Unsupervised learning:
  - Clustering to produce Grammars
  - Deep-Belief Networks (stacked RBMs)
- Supervised learning:
  - Training an MLP Melody Critic

# Some References

- Gillick, Tang, and Keller, **Machine Learning of Jazz Grammars**, *Computer Music Journal*, 34:3, pp. 56–66, Fall 2010, MIT Press.
- Bickerman, Bosley, Swire, and Keller, **Learning to Create Jazz Melodies Using Deep Belief Nets**, *Proc. First International Conference on Computational Creativity*, 228-237, January, 2010.
- Keller, Schofield, Toman-Yih, Merritt, **A Creative Improvisation Companion Base on Idiomatic Harmonic Bricks**, *Proc. Third ICCG*, June, 2012.
- Keller, Schofield, Toman-Yih, Merritt, and Elliott, **Automating the Explanation of Jazz Chord Progressions Using Idiomatic Analysis**, *Computer Music Journal*, 37:4, pp. 54–69, Winter 2013, MIT Press.