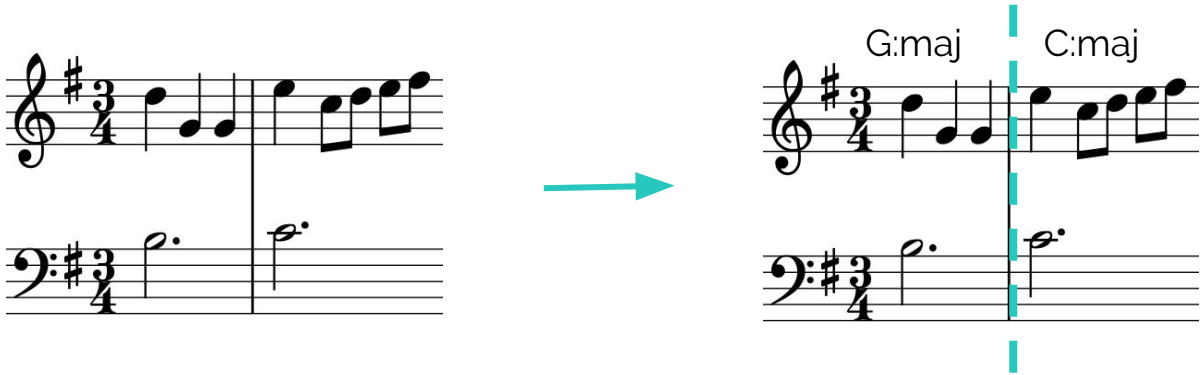
# Chord Recognition in Symbolic Music Using Semi-CRFs

Kristen Masada and Razvan Bunescu

# Quick Summary

We use machine learning to automatically detect chords in a piece of sheet music.

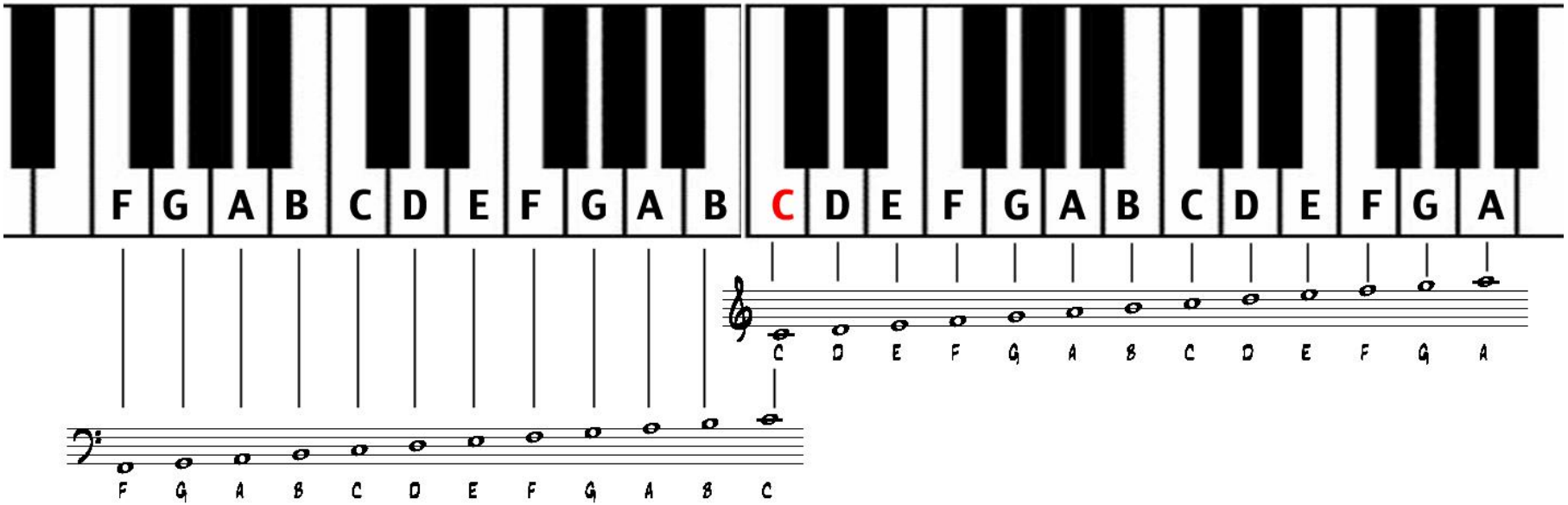



(Before)

(After)

# Music Theory Background

# How Do You Read Music?

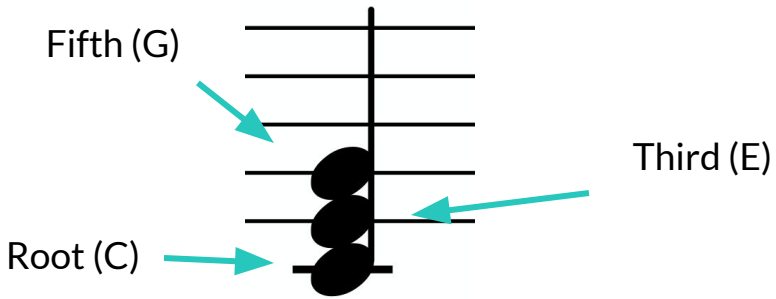



*Click on image above to open link to audio.

(image from musictheoryacademy.com)

# What Is a Chord?

- A group of 3 or more notes “that make sense when played or sung all at the same time” (*Harmony and Voice Leading*).
- **Triad** - Prototypical instance of a chord in Western tonal music.
  - 3 notes (**root**, **third**, and **fifth**).
  - *Why third and fifth?



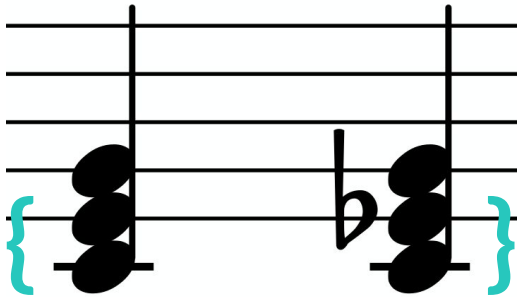

C chord

*Assume treble clef.

# What Is a Chord? (Cont.)

- Possesses a **mode**.
  - Determined by the **interval** (i.e. # of keys on piano) between the root and third.
  - E.g. major, minor.

Major third
(4 black/white keys)



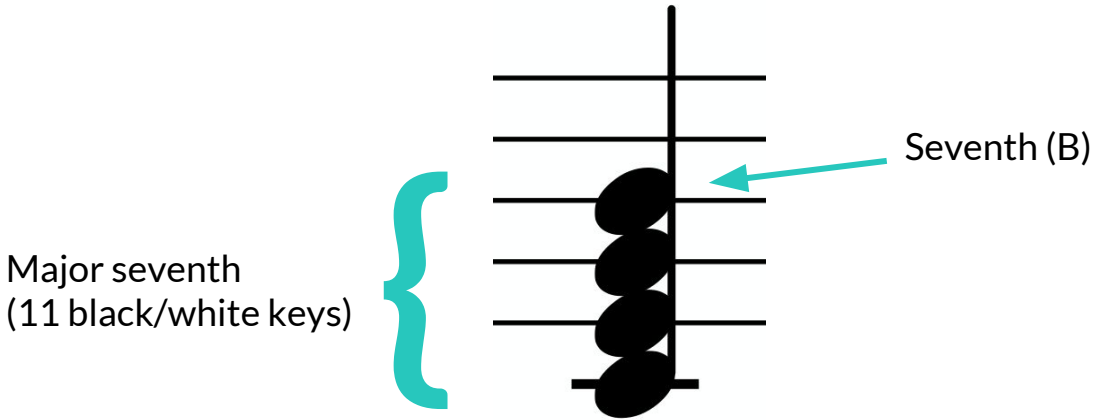
Minor third
(3 black/white keys)

C major C minor

*Click on chords above to open link to audio clip.

# What Is a Chord? (Cont.)

- Can have an **added note** (fourth note).
  - E.g. Fourth, sixth, seventh.

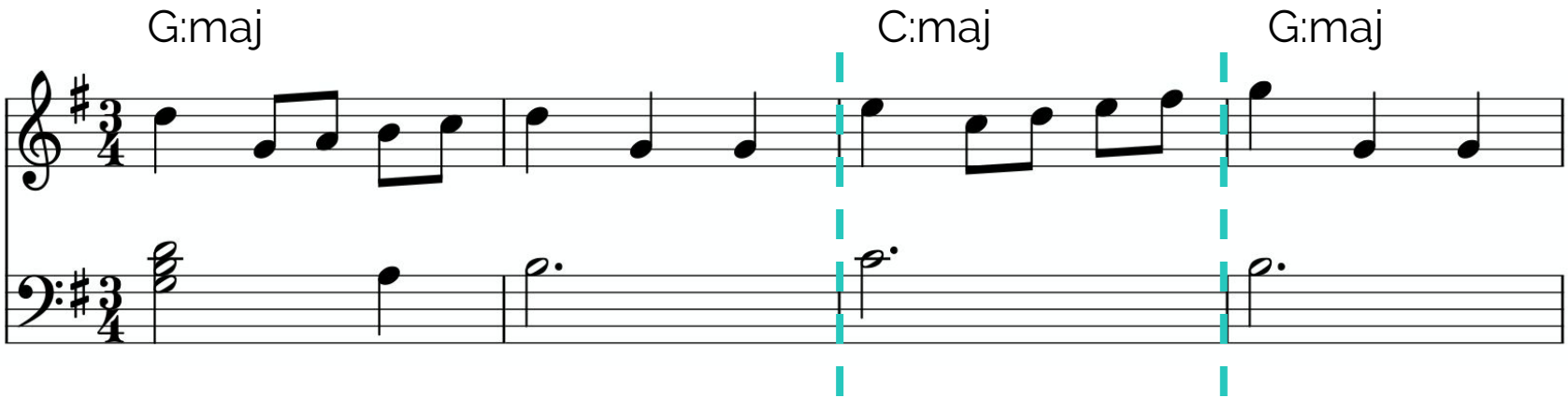



C major 7

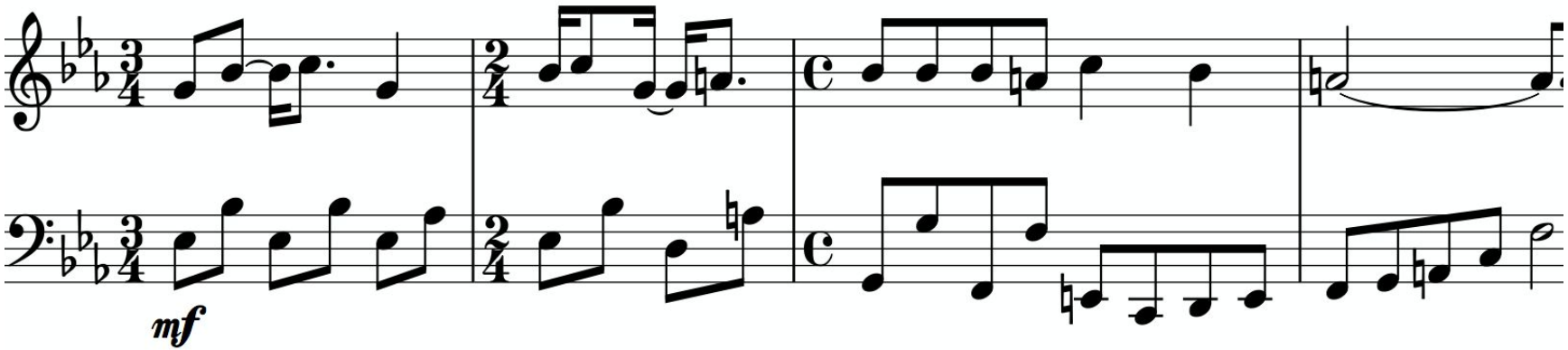
*Click on chord above to open link to audio.

# What Is Chord Recognition?

The partitioning of music into a sequence of segments such that the notes in each segment are consistent with a corresponding chord label.



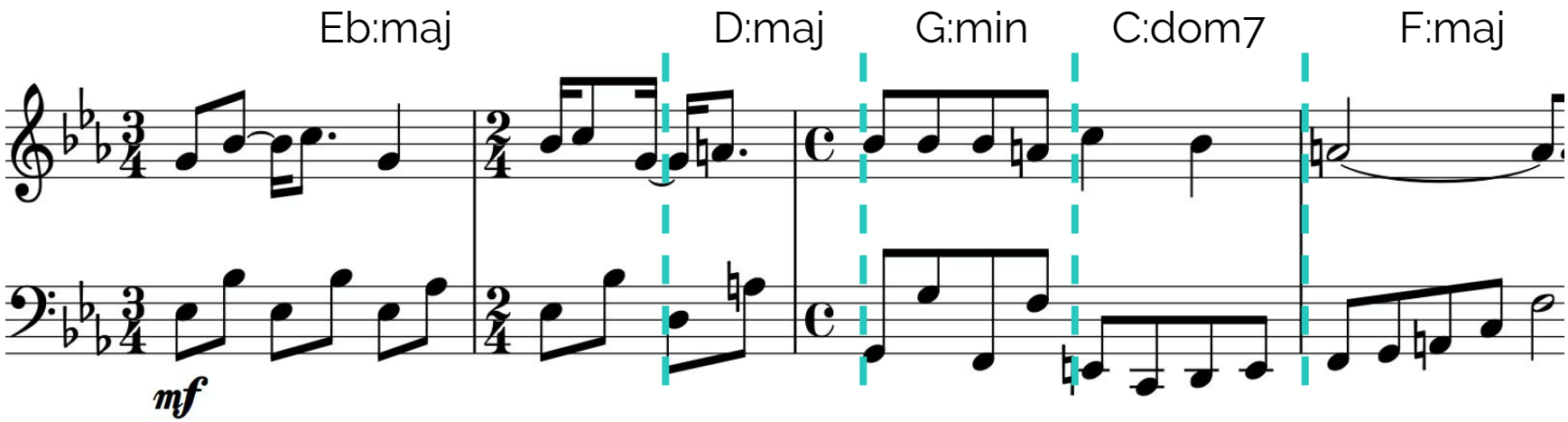
# “Martha My Dear” - the Beatles (Measures 1-4)




*Click on sheet music excerpt above to open link to audio.

# Which Chords Are Correct?

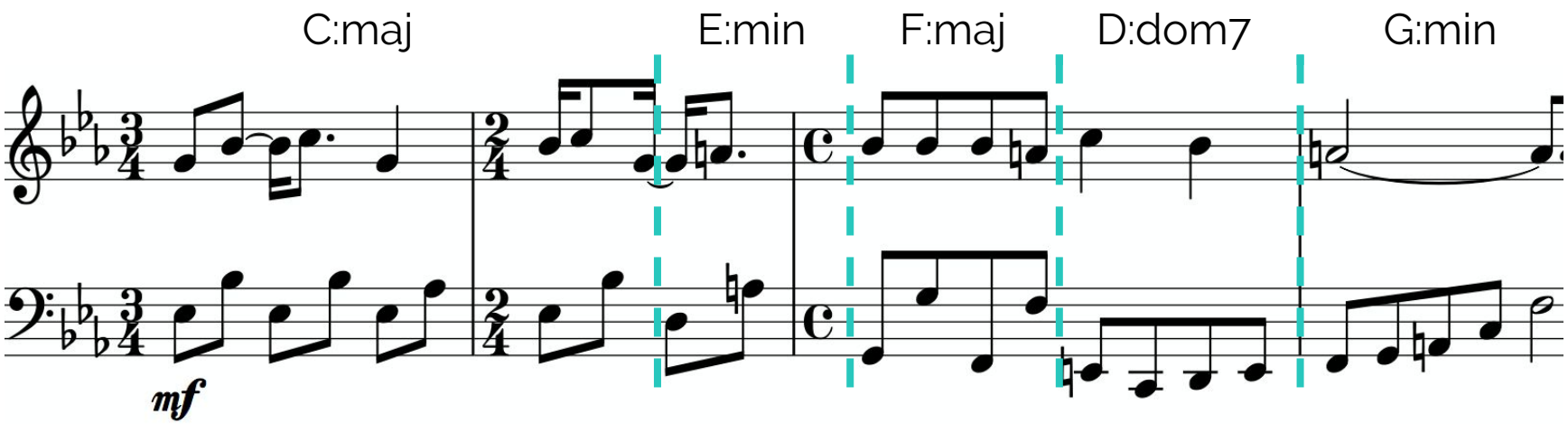
## Option 1:




*Click on sheet music excerpt above to open link to audio.

# Which Chords Are Correct? (Cont.)

## Option 2:

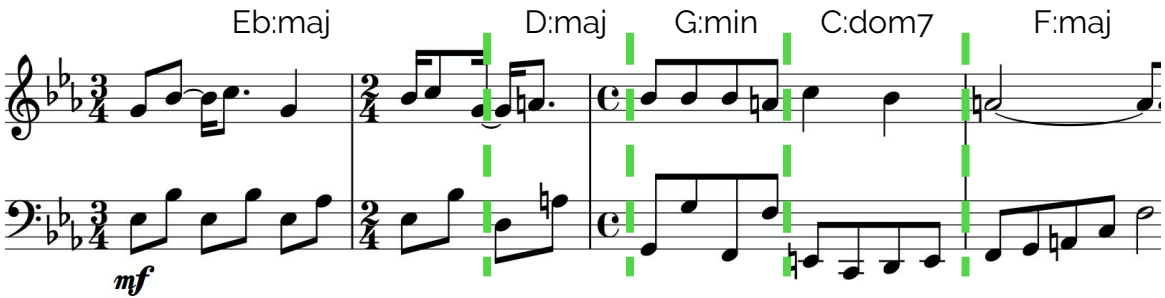



*Click on sheet music excerpt above to open link to audio.

# Option 1 Is Correct

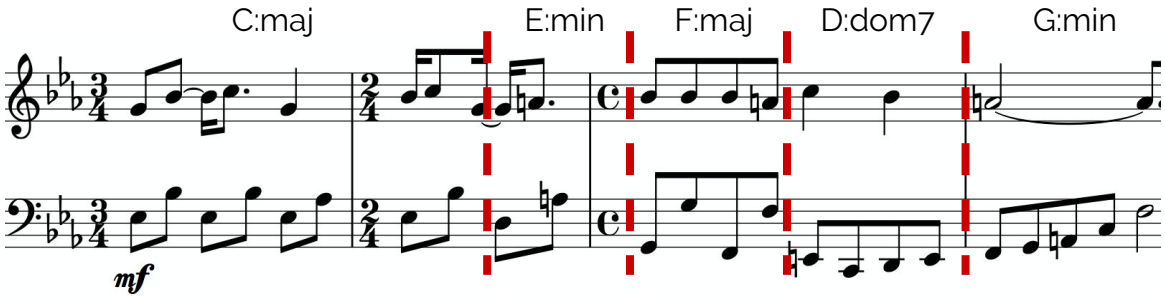
*Click on each sheet music excerpt to open link to audio.

## Option 1:




✓

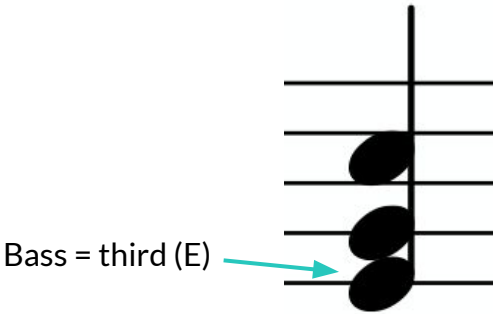
## Option 2:



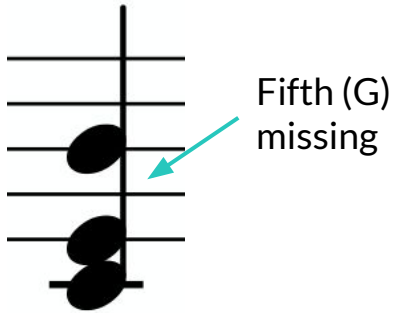

X

# Why Is Chord Recognition Hard?

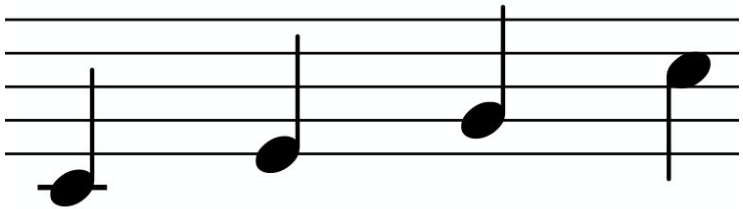
- Not as simple as identifying the notes in a segment
- Complications:
  - **Inversions** - root isn't bass note
  - Missing notes
  - Arpeggios - chord notes played one at a time




1st inversion
C major chord



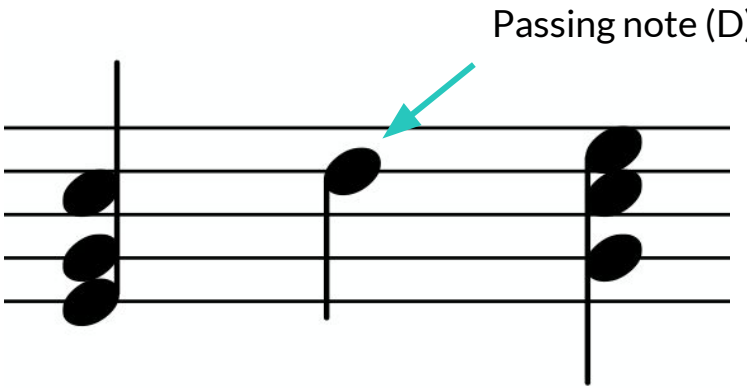

C major 7 chord
missing fifth



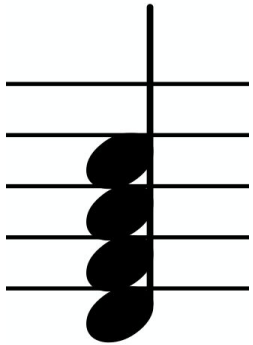
Arpeggiated C major chord

# Why Is Chord Recognition Hard? (Cont.)

- Complications (cont.):
  - **Non-harmonic tones** - notes that do not belong to any chord
    - E.g. **Passing tones**
  - Chords that share notes



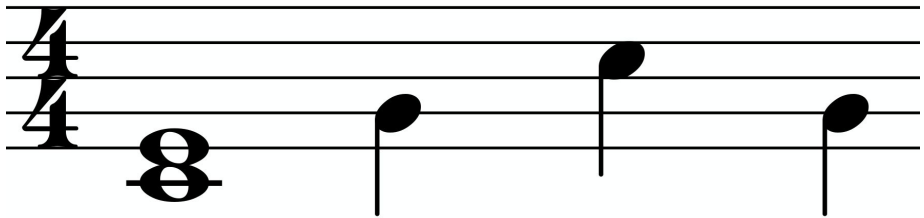

C major chord



D minor 7 chord or
F major 6 chord in third inversion

# Why Is Chord Recognition Hard? (Cont.)

- Which chord is “correct” for a given segment?
  - Duration
    - Root and third are usually long.
  - Accent - Perceptual prominence of beat.
    - Root and third more likely to occur on first (strongest) beat of measure.
  - Common chord progressions.



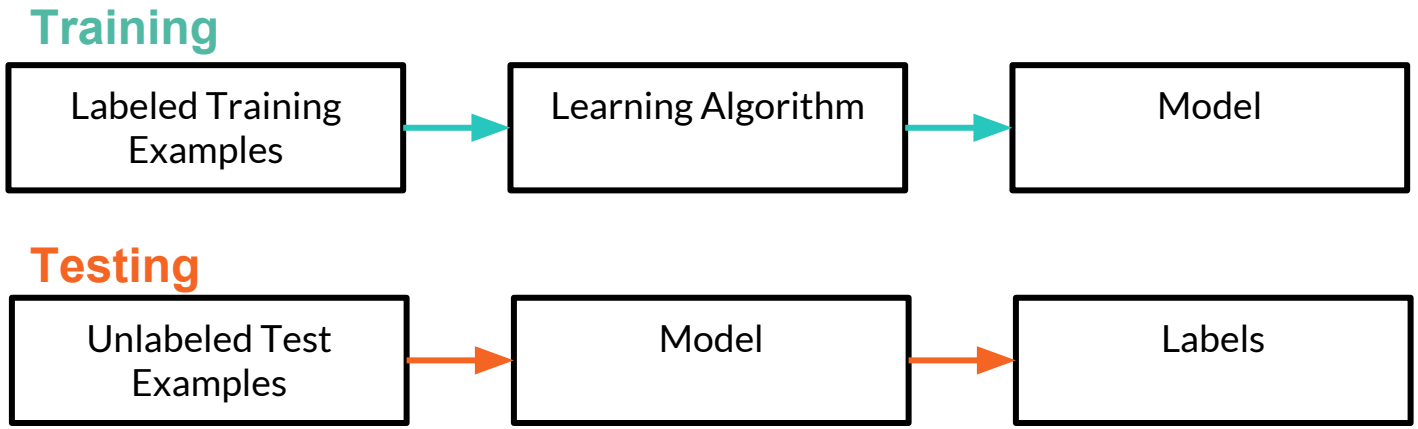
# Why Is Automatic Chord Recognition Useful?

- Music streaming services (e.g. Spotify, Apple Music)
  - Music identification
  - Music similarity computation
  - Automatic playlist generation
- Music information retrieval (MIR) research
  - Automatic accompaniment
  - Automatic composition
- Music education
- Playing instruments with other people

# Machine Learning + Semi-CRFs

# What Is Machine Learning?

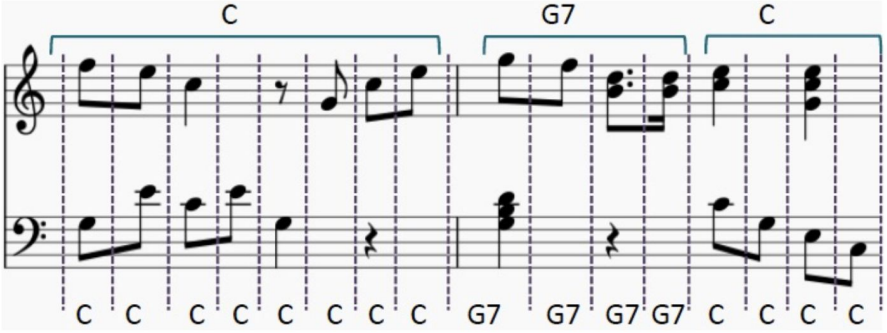
- “The science of getting computers to act without being explicitly programmed.” - Andrew Ng.
- **Supervised learning**




*Taken from Dr. Bunescu’s CS 4900, Lec 01 slides.

# Why Did We Choose Semi-CRFs?

- Radicioni and Esposito's "BREVE: An HMPerceptron-Based Chord Recognition System" (2010).
  - Breaks the musical input into **events** (each unique note onset/offset).
  - Trains a *Hidden Markov Model* (HMM) to assign a chord label to each event in the sequence.
- *BREVE can only exploit the characteristics of 1-2 events at a time, while semi-CRF can exploit entire segments.




Segments vs. Events

# What Are Semi-CRFs (semi-Markov Conditional Random Fields)?

- Partitions the given test example into segments and predict the labels that correspond to these segments (Sarawagi and Cohen, 2004).
- **s** = $\langle s_1, s_2, \ldots, s_K \rangle$ denotes a segmentation of the musical input **x**.
  - $s_k = \langle s_k.f, s_k.l \rangle$, where $s_k.f$ = index of first event, $s_k.l$ = index of last event.
- **y** = $\langle y_1, y_2, \ldots, y_K \rangle$ is the vector of chord labels corresponding to the segmentation ***s***.

$$
\begin{aligned}
P(\mathbf{s}, \mathbf{y} | \mathbf{x}, \mathbf{w}) &= \frac{e^{\mathbf{w}^T \mathbf{F}(\mathbf{s}, \mathbf{y}, \mathbf{x})}}{Z(\mathbf{x})} \\
Z(\mathbf{x}) &= \sum_{\mathbf{s}, \mathbf{y}} e^{\mathbf{w}^T \mathbf{F}(\mathbf{s}, \mathbf{y}, \mathbf{x})} \\
\mathbf{F}(\mathbf{s}, \mathbf{y}, \mathbf{x}) &= \sum_{k=1}^{K} \mathbf{f}(s_k, y_k, y_{k-1}, \mathbf{x})
\end{aligned}
$$

# What Are Semi-CRFs? (Cont.)

- We use Muis and Lu's semi-CRF package*.
  - Restricts segment features to 2 types:
    - Segment-label features - f($s_k$, $y_k$, **x**).
    - Transition features - g($y_k$, $y_{k-1}$, **x**).
    - *Allows for faster training/inference.

$$P(\mathbf{s}, \mathbf{y}|\mathbf{x}, \mathbf{w}) = \frac{e^{\mathbf{w}^T \mathbf{F}(\mathbf{s},\mathbf{y},\mathbf{x}) + \mathbf{u}^T \mathbf{G}(\mathbf{s},\mathbf{y},\mathbf{x})}}{Z(\mathbf{x})}$$

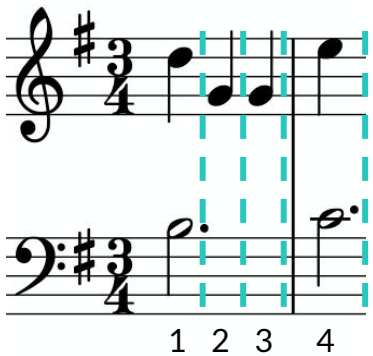
$$\mathbf{F}(\mathbf{s}, \mathbf{y}, \mathbf{x}) = \sum_{k=1}^{K} \mathbf{f}(s_k, y_k, \mathbf{x})$$

$$\mathbf{G}(\mathbf{s}, \mathbf{y}, \mathbf{x}) = \sum_{k=1}^{K} \mathbf{g}(y_k, y_{k-1}, \mathbf{x})$$

*http://statnlp.org/research/ie/

# How Do Semi-CRFs Work?

- Generates a **probabilistic graphical model**.
  - Each node is a candidate segment.
  - For each event, considers a number of candidate segments starting at that position.
    - Length 1, 2, up to a maximum segment length (usually 20).
- arg $\max_{s,y} P(\mathbf{s},\mathbf{y}|\mathbf{x},\mathbf{w})$ is computed using a semi-Markov analogue of the Viterbi algorithm.
  - Finds the most likely sequence of segments.



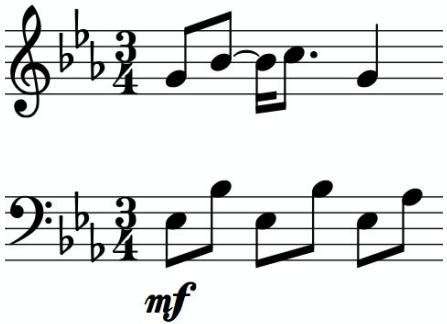

*y* = G:maj, *s.f* = 1, *s.l* = 2.
*y* = G:maj, *s.f* = 1, *s.l* = 3.
*y* = G:maj, *s.f* = 1, *s.l* = 4.
*y* = G:maj, *s.f* = 1, *s.l* = 5.

# Chord Recognition Features

# Segment Features - Segment Purity

- Feature $f_1$ determines the fraction of the notes in candidate segment *s* that are harmonic ( i.e. belong to chord *y*).

$$f_1(s, y) = \frac{\sum_{n \in s.Notes} \mathbf{1}[n \in y]}{|s.Notes|}$$




*y* = **Eb:maj**, *s* = (segment to left)
$f_1(s, y) = 9 / 10 = 90\%$*

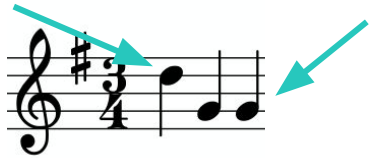
*Real-valued features are partitioned into the bin set [0.0, 0.1, ..., 0.9, 1.0] to convert them to boolean values.

# Segment Features - Chord Coverage

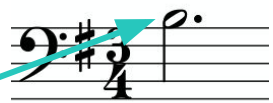
- Features $f_4$-$f_6$ determine which of the chord notes belongs to the candidate segment *s*.

$$
\begin{aligned}
f_4(s, y) &= \mathbf{1}[y.root \in s.Notes] \\
f_5(s, y) &= \mathbf{1}[y.third \in s.Notes] \\
f_6(s, y) &= \mathbf{1}[y.fifth \in s.Notes]
\end{aligned}
$$

*y.fifth* (D)



*y.root* (G)



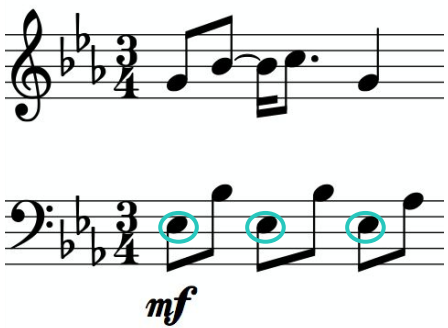
*y.third* (B)

*y* = G:maj, *s* = (segment to left)
$f_4(s, y) = 1$
$f_5(s, y) = 1$
$f_6(s, y) = 1$

# Segment Features - Chord Coverage (Cont.)

- Feature $f_{11}$ determines the fraction of the notes in candidate segment *s* that are the root note and weighs each note *n* by its duration *n.len*.

$$f_{11}(s, y) = \frac{\sum_{n \in s.Notes} \mathbf{1}[n = y.root] * n.len}{\sum_{n \in s.Notes} n.len}$$

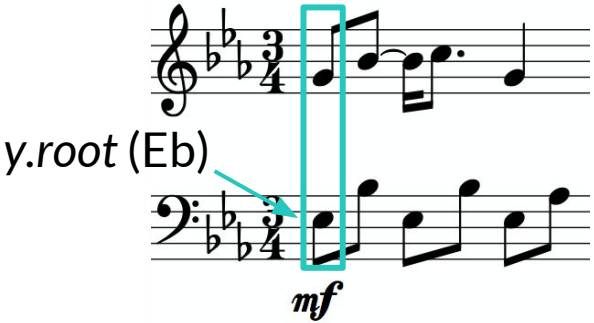



*y* = **Eb:maj**, *s* = (segment to left)
$f_{11}(s, y)$ = (1 * 0.5 + 1 * 0.5 + 1 * 0.5) / 6 = 25%

# Segment Features - Bass

- Features $f_{20}$-$f_{22}$ define the bass of segment *s* as the lowest note of the first event $e_1$ of *s*−−*s.e*$_1$*.bass*--and determines which chord note it corresponds to.

$$
\begin{aligned}
f_{20}(s, y) &= \mathbf{1}[s.e_1.bass = y.root] \\
f_{21}(s, y) &= \mathbf{1}[s.e_1.bass = y.third] \\
f_{22}(s, y) &= \mathbf{1}[s.e_1.bass = y.fifth]
\end{aligned}
$$

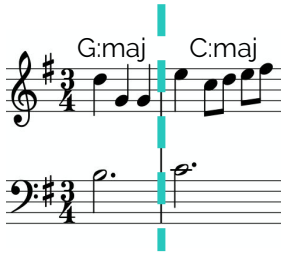



*y* = Eb:maj, *s* = (segment to left)
$f_{20}(s, y) = 1$
$f_{21}(s, y) = 0$
$f_{22}(s, y) = 0$

# Transition Features - Chord Bigram

- Feature template *g1* captures chord transition information between the current segment *y* and the previous segment *y'*.
  - *mode*: major (M), minor (m), or diminished (d).
  - *added* note: none (∅), fourth (4), sixth (6), or seventh (7).
  - interval in semitones between the roots of the two chords.
    - *generalizes to unseen chords.
  - *Generates 1,728 possible features.

$$g_1(y, y') = \mathbf{1}[y.mode = \{M, m, d\} \wedge y.added = \{\emptyset, 4, 6, 7\} \wedge$$
$$y'.mode = \{M, m, d\} \wedge y'.added = \{\emptyset, 4, 6, 7\} \wedge$$
$$|y.root - y'.root| = \{0, 1, ..., 11\}]$$




*y* =C:maj, *y'* =G:maj

$g_1(y, y')$ = 1 [ M ^ ∅ ^ M ^ ∅ ^ 7] = 1

# Datasets

# +

# Experimental Evaluation

# Bach Choral Harmony Dataset (BaCh)

- Corpus of 60 four-part Bach chorales.
  - Contains 5,664 events and 3,090 segments.
- **10 fold cross-validation**.
  - 60 chorales / 10 folds = 6 songs per fold.
  - 9 folds for training, 1 for testing.
- 102 chords in dataset.
  - 3 modes (major, minor, diminished) x 12 root pitches x 4 added notes (none, 4, 6, 7) = 144 total possible labels.
- *Features that appear less than 5 times in training removed.

| System | $Acc_E$ | $P_S$ | $R_S$ | $F_S$ |
|---|---|---|---|---|
| semi-CRF | **83.16%** | **77.60%** | **73.48%** | **75.48%** |
| HMP-tron* | 80.30% | 74.18% | 69.76% | 71.90% |

# TAVERN

- Corpus of 27 complete sets of theme and variations for piano, composed by Mozart and Beethoven.
  - Contains 63,876 events and 12,802 segments.
- One split.
  - Training - 11 Beethoven + 6 Mozart sets.
  - Testing - 6 Beethoven + 4 Mozart sets.
- 69 chords in dataset.

| System | $Acc_E$ | $P_S$ | $R_S$ | $F_S$ |
|---|---|---|---|---|
| semi-CRF | **77.47%** | **66.86%** | **60.35%** | **63.44%** |
| HMP-tron* | 60.55% | 27.83% | 23.21% | 25.31% |

# Questions

Our research paper: http://bit.ly/2jxZ0Dg