Contextual Chord Embeddings for Stylistic Analysis
Matt Chiu, Eastman School of Music
Presentation link: https://youtu.be/iSHihr-M7-U

Table of Contents
Code
Definitions
Corpora
Word2vec parameters
Mozart model
Rock model
Word2vec mathematics
Acknowledgements
References

Code
https://www.shrunken.com/ChordEmbeddingsCode

Definitions
*Computation hermeneutics:* Encoding meaning into a computer
*Class representation:* A class describes a group of objects all belonging to it
*Contextual representation:* Objects gain associated meaning within particular contexts
*Stylistic representation:* The stylistic context defining contextual and class representations
*Vector:* Lists of numbers where there are as many dimensions as the vector is long
*Word embedding:* Vector representations for words made of real numbers
*Dendrogram:* Hierarchical clustering method
*Euclidean distance:* Distance between two vectors $\mathbf{u}$ and $\mathbf{v}$ defined as
\[
d(\mathbf{u}, \mathbf{v}) = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \cdots + (u_n - v_n)^2}
\]
*Cosine similarity:* Similarity between two vectors $\mathbf{U}$ and $\mathbf{V}$ defined as
\[
\cos \theta = \frac{\sum u_i v_i}{\sqrt{\sum u_i^2} \sqrt{\sum v_i^2}}
\]
*Cosine distance:* Distance between two vectors $\mathbf{u}$ and $\mathbf{v}$ defined as $1 - \cos \theta$.

Corpora
*Yale Classical Archives corpus* (White and Quinn 2016)
*Rolling Stone corpus* (Temperley and de Clercq 2011/2013)
**Word2vec parameters:**
There are a variety of parameters when using the word2vec algorithm. These are the ones that this paper makes use of:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corpus</td>
<td>Corpus being parsed</td>
</tr>
<tr>
<td>Window size</td>
<td>Maximum distance between target and contexts</td>
</tr>
<tr>
<td>Vector size</td>
<td>Vector dimensionality</td>
</tr>
<tr>
<td>Skip-gram</td>
<td>({1,0}) — whether using skip-gram or continuous bag of words</td>
</tr>
<tr>
<td>Epochs</td>
<td>Number of iterations through the corpus</td>
</tr>
</tbody>
</table>

**Mozart model**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corpus</td>
<td>YCAC, Mozart file</td>
</tr>
<tr>
<td>Window size</td>
<td>4</td>
</tr>
<tr>
<td>Vector size</td>
<td>30</td>
</tr>
<tr>
<td>Skip-gram</td>
<td>1 : Skip-gram</td>
</tr>
<tr>
<td>Epochs</td>
<td>40</td>
</tr>
</tbody>
</table>

-Mozart model: dendrogram with key distances

-Diatonic similarity to C-major chord embedding

-TPS and Chord Embeddings: Measured distance from C major

\[ r = .921 \quad p-value = .009 \]
## Rock Model

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Rolling Stone corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window size</td>
<td>2</td>
</tr>
<tr>
<td>Vector size</td>
<td>20</td>
</tr>
<tr>
<td>Skip-gram</td>
<td>1 : Skip-gram</td>
</tr>
<tr>
<td>Epochs</td>
<td>100</td>
</tr>
</tbody>
</table>

### Rock Model: diatonic similarity to C-major chord embedding

![Graph showing diatonic similarity to C-major chord embedding](image)

### Chord Position

![Graph showing adjacent chord distance (cosine)](image)
Word2vec mathematical details

I’ve attempted to condense core principles of word2vec, but anyone interested in investigating further should look to the genism documentation page, or Jurafsky and Martin (2020).

Probability
The probability of the target word \( t \) occurring near a context word \( c \) is calculated by taking the dot product between the vectors and turning the dot product into a probability with the sigmoid function.

Given a target and context tuple, the probability that the context word is real: \( P(+/t,c) \)

Dot product: \( \text{Similarity}(t,c) \approx t \cdot c \)

Sigmoid function: \( \sigma(x) = \frac{1}{1+e^{-x}} \)

To model the probabilities within the window of \( L \) chords:
\[ P(+|t, c) = \sum_{i=1}^{L} \log \sigma(t \cdot c_i) = \frac{1}{1 + e^{-t \cdot c_i}} \]

**Updating embeddings**

Embedding are changed to maximize the dot product between the target chord and chords that appear near it \(c_{pos}\). We also want to minimize the dot product between negative samples (chords that don’t occur near one another) \(c_{neg}\). The amount of negative samples is specified \(k\). To change the embeddings, word2vec uses gradient descent—an operation for finding the minimum of a differentiable function. It therefore requires the derivative of a loss function \(E\), modeled as cross entropy.

\[
E_{CE} = - \left[ \log P(+|t, c_{pos}) + \sum_{i}^{k} \log P(-|t, c_{neg}) \right]
\]

\[
= - \left[ \log \sigma(\cdot c_{pos}) + \sum_{i}^{k} \log \sigma(\cdot c_{neg}) \right]
\]

Iterating through the corpus, the embeddings are updated from chord \(n\) to \(n+1\). Where \(\eta\) is the “learning rate” or the step size for each iteration while moving towards the local minimum.

\[
t^{n+1} = t^n - \eta [\sigma(c_{pos} \cdot t^n) - 1]c_{pos} + \sum_{i=1}^{k} [\sigma(c_{neg} \cdot t^n)c_{neg}]i
\]

**Acknowledgements**

Diane Litman for introducing me to the field of natural language processing
Eron Smith and Davy Temperley for feedback and revisions
Noah Kahrs for exchanging code and working out details at the projects incipit

I’d also like to acknowledge that I’m recording from Rochester, NY—and from unceded land of the Seneca peoples. I ask you to join me in recognizing the lands founded upon exclusion and erasure of many indigenous peoples, and to acknowledge the Seneca community, past present and future generations.
References


