A NEURAL GREEDY MODEL FOR VOICE SEPARATION IN SYMBOLIC MUSIC

Patrick Gray
School of EECS
Ohio University, Athens, OH
pg219709@ohio.edu

Razvan Bunescu
School of EECS
Ohio University, Athens, OH
bunescu@ohio.edu

ABSTRACT

Music is often experienced as a simultaneous progression of multiple streams of notes, or voices. The automatic separation of music into voices is complicated by the fact that music spans a voice-leading continuum ranging from monophonic, to homophonic, to polyphonic, often within the same work. We address this diversity by defining voice separation as the task of partitioning music into streams that exhibit both a high degree of external perceptual separation from the other streams and a high degree of internal perceptual consistency, to the maximum degree that is possible in the given musical input. Equipped with this task definition, we manually annotated a corpus of popular music and used it to train a neural network with one hidden layer that is connected to a diverse set of perceptually informed input features. The trained neural model greedily assigns notes to voices in a left to right traversal of the input chord sequence. When evaluated on the extraction of consecutive within voice note pairs, the model obtains over 91% F-measure, surpassing a strong baseline based on an iterative application of an envelope extraction function.

1. INTRODUCTION AND MOTIVATION

The separation of symbolic music into perceptually independent streams of notes, i.e. voices or lines, is generally considered to be an important pre-processing step for a number of applications in music information retrieval, such as query by humming (matching monophonic queries against databases of polyphonic or homophonic music) [13] or theme identification [12]. Voice separation adds structure to music and thus enables the implementation of more sophisticated music analysis tasks [17]. Depending on their definition of voice, existing approaches to voice separation in symbolic music can be organized in two main categories: 1) approaches that extract voices as monophonic sequences of successive non-overlapping musical notes [5, 6, 8, 11, 14, 16, 17]; and 2) approaches that allow voices to contain simultaneous note events, such as chords [4, 9, 10, 15]. Approaches in the first category typically use the musicological notion of voice that is referenced in the voice-leading rules of the Western musical tradition, rules that govern the horizontal motion of individual voices from note to note in successive chords [1, 4]. Starting with [4], approaches in the second category break with the musicological notion of voice and emphasize a perceptual view of musical voice that corresponds more closely to the notion of independent auditory streams [2, 3]. Orthogonal to this categorization, a limited number of voice separation approaches are formulated as parametric models, with parameters that are trained on music already labeled with voice information [6, 8, 11].

In this paper, we propose a data-driven approach to voice separation that preserves the musicological notion of voice. Our aim is to obtain a segregation of music into voices that would enable a downstream system to determine whether an arbitrary musical input satisfies the known set of voice-leading rules, or conversely identify places where the input violates voice-leading rules.

2. TASK DEFINITION

According to Huron [7], "the principal purpose of voice-leading is to create perceptually independent musical lines". However, if a voice is taken to be a monophonic sequence of notes, as implied by traditional voice-leading rules [1], then not all music is composed of independent musical lines. In homophonic accompaniment, for example, multiple musical lines (are meant to) fuse together into one perceptual stream. As Cambouropoulos [4] observes for homophonic accompaniment, "traditional voice-leading results in perceivable musical *texture*, not independent musical lines". In contrast with the traditional notion of voice used in previous voice separation approaches, Cambouropoulos redefines in [4] the task of 'voice' separation as that of separating music into perceptually independent musical *streams*, where a stream may contain two or more synchronous notes that are perceived as fusing in the same auditory stream. This definition is used in [9, 15] to build automatic approaches for splitting symbolic music into perceptually independent musical streams.

Since a major aim of our approach is to enable building "musical critics" that automatically determine whether an arbitrary musical input obeys traditional voice-leading rules, we adopt the musicological notion of voice as a

monophonic sequence of non-overlapping notes. This definition however leads to an underspecified voice separation task: for any non-trivial musical input, there usually is a large number of possible separations into voices that satisfy the constraints that they are monophonic and contain notes in chronological order that do not overlap. Further constraining the voices to be perceptually independent would mean the definition could no longer apply to music with homophonic textures, as Cambouropoulos correctly noticed in [4]. Since we intend the voice separation approach to be applicable to arbitrary musical input, we instead define voice separation as follows:

Definition 1. *Voice separation is the task of partitioning music into monophonic sequences (voices) of non-overlapping notes that exhibit both a high degree of external perceptual separation from the other voices and a high degree of internal perceptual consistency, to the maximum degree that is possible in the given musical input.*

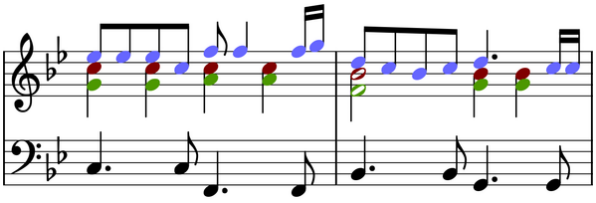

Figure 1. Example voice separation from "Earth Song".

Figure 1 shows a simple example of voice separation obtained using the definition above. While the soprano and bass lines can be heard as perceptually distinct voices, we cannot say the same for the tenor and alto lines shown in green and red, respectively. However, clear perceptual independence is not needed under the new task definition. The two intermediate voices exhibit a high degree of perceptual consistency: their consecutive notes satisfy to a large extent the pitch proximity and temporal continuity principles needed to evoke strong auditory streams [7]. Indeed, when heard in isolation, both the tenor and the alto are heard as continuous auditory streams, the same streams that are also heard when the two are played together. The two streams do not overlap, which helps with perceptual tracking [7]. Furthermore, out of all the streaming possibilities, they also exhibit the largest possible degree of external perceptual separation from each other and from the other voices in the given musical input.

3. ANNOTATION GUIDELINES

According to the definition in Section 2, voice separation requires partitioning music into monophonic sequences of non-overlapping notes that exhibit a high degree of perceptual salience, to the maximum extent that is possible in the given musical input. As such, an overriding principle that we followed during the manual annotation process was to always give precedence to what was heard in the music, even when this appeared to contradict formal perceptual principles, such as pitch proximity. Furthermore, whenever formal principles seemed to be violated by perceptual streams, an attempt was made to explain the apparent conflict. Providing justifications for non-trivial annotation decisions enabled refining existing formal perceptual principles and also informed the feature engineering effort.

Listening to the original music is often not sufficient on its own for voice separation, as not all the voices contained in a given musical input can be distinctly heard. Because we give precedence to perception, we first annotated those voices that could be distinguished clearly in the music, which often meant annotating first the melodic lines in the soprano and the bass. When the intermediate voices were difficult to hear because of being masked by more salient voices, one simple test was to remove the already annotated most prominent voice (often the soprano [1]) and listen to the result. Alternatively, when multiple conflicting voice separations were plausible, we annotated the voice that, after listening to it in isolation, was easiest to distinguish perceptually in the original music.

Figure 2 shows two examples where the perceptual principle of pitch proximity appears to conflict with what is heard as the most salient voice. In the first measure, the first D_4 note can continue with any of the 3 notes in the following I^6 chord. However, although the bass note in the chord has the same pitch, we hear the first D_4 most saliently as part of the melody in the soprano. The D_4 can also be heard as creating a musical line with the next D_4 notes in the bass, although less prominently. The least salient voice assignment would be between the D_4 and the intermediate line that starts on the following G_4. While we annotate all these streaming possibilities (as shown in Figure 7), we mark the soprano line assignment as the most salient for the D_4. Similarly, in the last chord from the second measure from Figure 2, although E_4 is closer to the previous F_4, it is the G_4 that is most prominently heard as continuing the soprano line. This was likely reinforced by the fact that the G_4 in the last chord was "prepared" by the G_4 preceding F_4.

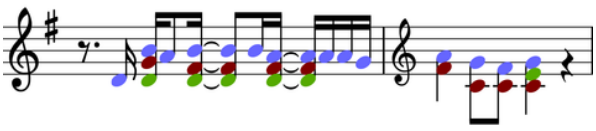

Figure 2. Voice separation annotations, for measures 5 in "Knockin' on Heaven's Door" and 12 in "Let It Be".

Other non-trivial annotation decisions, especially in the beginning of the annotation effort, involved whether two streams should be connected or not. Overall, we adopted the guideline that we should break the music into fewer and consequently longer voices, especially if validated perceptually. Figure 3, for example, shows the A_4 in the third measure connected to the following C_5. Even though the two notes are separated by a quarter rest, they are heard as belonging to the same stream, which may also be helped by the relatively long duration of A_4 and by the fact that the same pattern is repeated in the piece. We have also dis-

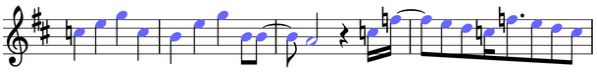

Figure 3. Voice separation annotation in the treble for measures 38-41 in "Count on Me".

covered that "preparation" through previous occurrences of the same note or notes one octave above or below can significantly attenuate the effect of a large pitch distance and thus help with connecting the note to an active stream. This effect is shown in Figure 4, where the voice in the first measure is most prominently heard as continuing with the B_4 in the second measure.

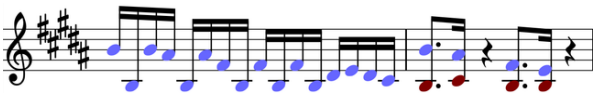

Figure 4. Voice separation annotation in the treble for measures 26-27 in "A Thousand Miles".

Sometimes, the assignment of a note to one of the available active voices is hard to make due to inherent musical ambiguity. An example is shown in Figure 5, where it is hard to determine if the A_4 in the second measure connects to the top C_6 or the C_5 one octave below. After being played separately, each voice assignment can be distinguished perceptually in the original music. The C_5 is closer in pitch to the A_4 and it is also in a range with better defined pitch sensations than the C_6. On the other hand, the pitch distance between the upper C_6 and the A_4 is attenuated by the synchronous C_5. Eventually we annotated A_4 as connecting to the slightly more salient C_5, but also marked it as ambiguous between the two C notes.

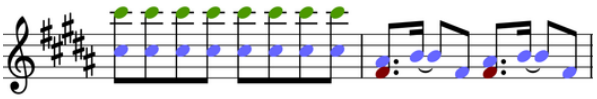

Figure 5. Voice separation annotation in the treble for measures 62-63 in "A Thousand Miles".

Other examples of harmony influencing voice assignment involve the seventh scale degree notes appearing in VII and VII^6 chords. As shown in Figure 6, when such a chord is first used, the $\hat{7}$ note does connect to any of the previous streams, despite the closer pitch proximity.

4. VOICE SEPARATION DATASET

We compiled a corpus of piano versions of 20 popular compositions of varying complexity that are representative of many genres of music. Each song was downloaded from *www.musescore.com* and converted to MusicXML. In selecting music, we followed a few basic criteria. First, we avoided collecting piano accompaniments and gave preference to piano renditions that sounded as much as possible like the original song. Among other things, this ensured that each score contained at least one clearly defined

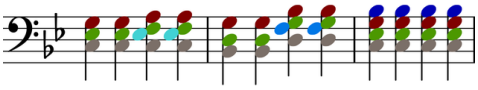

Figure 6. Voice separation annotation in the bass for measures 26-28 in "Earth Song".

melody. Second, we collected only tonal music. Atonal music is often comprised of unusual melodic structures, which were observed to lead to a poor perception of voices by the annotators. Following the annotation guidelines, we manually labeled the voice for each note in the dataset. The annotations will be made publicly available. The names of the 20 musical pieces are shown in Table 1, together with statistics such as the total number of notes, number of voices, average number of notes per voice, number of within-voice note pairs, number of unique note onsets, and average number of notes per chord. The 20 songs were manually annotated by the first author; additionally, the 10 songs marked with a star were also annotated by the second author. In terms of F-measure, the inter-annotator agreement (ITA) on the 10 songs is 96.08% (more detailed ITA numbers are shown in Table 2). The last column shows the (macro-averaged) F-measure of our neural greedy model, to be discussed in Section 6. As can be seen in Table 1, the number of voices varies widely, ranging between 4 for Greensleeves to 123 for 21 Guns, the longest musical composition, with a variable musical texture and frequent breaks in the harmonic accompaniment of the melody. The last line shows the same total/average statistics for the first 50 four-part Bach Chorales available in Music21, for which we use the original partition into voices, without the duplication of unisons.

5. THE VOICE SEPARATION MODEL

To separate a musical input into its constituent voices, we first order all the notes based on their onsets into a sequence of *chords* $\mathcal{C} = \{c_1, c_2, ..., c_T\}$, where a chord is defined to be a maximal group of notes that have the same onset. Assignment of notes to voices is then performed in chronological order, from left to right, starting with the first chord c_1. Because voices are by definition monophonic, each note in the first chord is considered to start a separate, new voice. These first voices, together with an empty voice ϵ, constitute the initial set of *active voices* $\mathcal{V}$. At each onset t, the algorithm greedily assigns a note n from the current chord c_t to one of the voices in the active set by selecting the active voice v that maximizes a trained assignment probability $p(n, v)$, i.e. $v(n) = \arg\max_{v \in \hat{\mathcal{V}}} p(n, v)$. Notes from the current chord are assigned to voices in the order of their maximal score $p(n, v(n))$. If a note is assigned to the empty voice, then a new voice is added to the active set. The set of candidate active voices $\hat{\mathcal{V}}$ available for any given note n is a subset of active voices $\mathcal{V}$ constrained such that assigning n to any of the voices in $\hat{\mathcal{V}}$ would not lead to crossing voices or to multiple synchronous notes being assigned to the same voice.

Popular Music dataset	# Notes	# Voices	# N / V	# Pairs	# Onsets	Synchronicity	F-measure
21 Guns (Green Day)	1969	123	16.01	1801	666	2.96	86.24
Apples to the Core (Daniel Ingram)	923	29	31.83	892	397	2.32	77.67
Count on Me (Bruno Mars)	775	11	70.45	764	473	1.64	97.22
Dreams (Rogue)*	615	12	51.25	603	474	1.30	98.32
Earth Song (Michael Jackson)*	431	15	28.73	416	216	2.00	93.27
Endless Love (Lionel Richie)	909	23	39.52	886	481	1.89	96.52
Forest (Twenty One Pilots)	1784	89	20.04	1695	1090	1.64	91.93
Fur Elise (Ludwig van Beethoven)*	900	77	11.69	823	653	1.38	91.98
Greensleeves*	231	4	57.75	213	72	3.21	92.88
How to Save a Life (The Fray)*	440	13	33.85	427	291	1.51	98.11
Hymn for the Weekend (Coldplay)	1269	50	25.38	1218	706	1.80	92.30
Knockin' on Heaven's Door (Bob Dylan)*	355	41	8.66	312	180	1.97	90.92
Let It Be (The Beatles)*	563	22	25.59	540	251	2.24	87.29
One Call Away (Charlie Puth)	993	56	17.73	937	505	1.97	91.33
See You Again (Wiz Khalifa)*	704	66	10.67	638	359	1.96	81.16
Teenagers (My Chemical Romance)	315	18	17.50	297	145	2.17	91.39
A Thousand Miles (Vanessa Carlton)*	1001	61	16.41	937	458	2.19	96.61
To a Wild Rose (Edward Macdowell)	307	20	15.35	287	132	2.33	88.72
Uptown Girl (Billy Joel)	606	46	13.17	560	297	2.04	93.41
When I Look at You (Miley Cyrus)*	1152	82	14.05	1067	683	1.69	92.92
Totals & Averages	16242	42.90	26.28	15313	8529	2.01	91.51
Bach Chorales dataset	12282	4	61.41	11874	4519	2.73	95.47

Table 1. Statistics for the Popular Music dataset and the Bach Chorales dataset.

The assignment probability $p(n, v)$ captures the compatibility between a note n and an active voice v. To compute it, we first define a vector $\Phi(n, v)$ of perceptually informed compatibility features (Section 5.2). The probability is then computed as $p(n, v) = \sigma(\mathbf{w}^T h_W(n, v))$, where σ is the sigmoid function and $h_W(n, v)$ is the vector of activations of the neurons on the last (hidden) layer in a neural network with input $\Phi(n, v)$.

To train the network parameters $\theta = [\mathbf{w}, W]$, we maximize the likelihood of the training data:

$$\hat{\theta} = \arg\max_{\theta} \prod_{t=1}^{T} \prod_{n \in c_t} \prod_{v \in \hat{\mathcal{V}}} p(n,v|\theta)^{l(n,v)} (1-p(n,v|\theta))^{1-l(n,v)} \quad (1)$$

where $l(n, v)$ is a binary label that indicates whether or not note n was annotated to belong to voice v in the training data. This formulation of the objective function is flexible enough to be used in 2 types of voice separation scenarios:

1. **Ranking**: Assign a note to the top-ranked candidate active voice, i.e. $v(n) = \arg\max_{v \in \hat{\mathcal{V}}} p(n, v)$.
2. **Multi-label classification**: Assign a note to all candidate active voices whose assignment probability is large enough, i.e. $V(n) = \{v \in \hat{\mathcal{V}} | p(n, v) > 0.5\}$.

The first scenario is the simplest one and rests on the working assumption that a note can belong to a single voice. The second scenario is more general and allows a note to belong to more than one voice. Such capability would be useful in cases where a note is heard simultaneously as part of two musical streams. Figure 7, for example, shows the voice separation performed under the two scenarios for the same measure. In the ranking approach shown on the left, we label the second F_4 as belonging to the soprano voice. Since in this scenario we can assign a note to just one voice, we select the voice assignment that is heard as the most salient, which in this case is the soprano. In the multi-label approach shown on the right, we label the second F_4 as belonging to both active voices, since the note is heard as belonging to both. In the experiments that we report in this paper (Section 6), we used the simpler ranking approach, leaving the more general multi-label approach for future work.

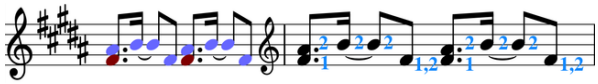

Figure 7. Two voice separation scenarios, for measure 16 from "A Thousand Miles".

5.1 Iterative Envelope Extraction

We also propose a baseline system for voice-separation that iteratively extracts the upper *envelope* i.e. the topmost monophonic sequence of non-overlapping notes. Figure 8 shows how the iterative envelope extraction process works on the second measure from Figure 2, copied here for readability. The top left measure is the original measure from Figure 2 and we use it as the current input. Its upper envelope is shown in the bottom left measure, which will become the first voice. After extracting the first voice from the input, we obtain the second measure in the top staff, which is now set to be the current input. We again apply

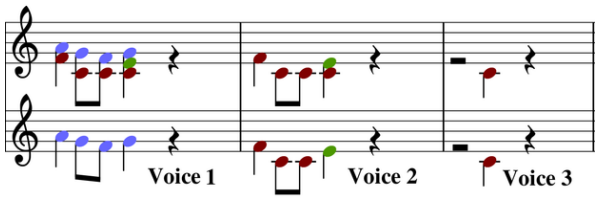

Figure 8. Voice separation as iterative envelope extraction.

the same envelope extraction process to obtain the second voice, shown in the second measure on the bottom staff. After extracting the second voice from the current input, we obtain a new current input, shown in the third measure on the top staff. Extracting the third voice from the current input results in an empty set and correspondingly the baseline algorithm stops. For this input, the baseline extracted voice 1 without errors, however it made a mistake in the last note assignment for voice 2.

5.2 Voice Separation Features

The assignment probability $p(n, v)$ is computed by the neural model based on a vector of input features $\Phi(n, v) = [\phi_0, \phi_1, ..., \phi_K]$ that will be described in this section, using $v.last$ to denote the last note in the active voice v.

5.2.1 Empty Voice Feature

The empty voice feature ϕ_0 is set to 1 only for the empty voice, i.e. $\phi_0(n, \epsilon) = 1$ and $\phi_0(n, v) = 0, \forall v \neq \epsilon$. All the remaining features in any feature vector for an empty voice $\Phi(n, \epsilon)$ are set to zero. This allows the empty voice to activate a bias parameter w_0, which is equivalent to learning a threshold $-w_0$ that the weighted combination of the remaining features must exceed in order for the note to be assigned to an existing, non-empty, active voice. Otherwise, the note n will be assigned to the empty voice, meaning it will start a new voice.

5.2.2 Pitch and Pitch Proximity Features

According to Huron's formulation of the pitch proximity principle, *the coherence of an auditory stream is maintained by close pitch proximity in successive tones within the stream* [7]. Correspondingly, we define a pitch proximity feature $\phi_1(n, v) = pd(n, v.last) = |ps(n) - ps(v.last)|$ to be the absolute distance in half steps between the pitch space representations of notes n and $v.last$. The pitch proximity feature enables our system to quickly learn that notes rarely pair with voices lying at intervals beyond an octave. We also add two features $\phi_2(n, v) = ps(n)$ and $\phi_3(n, v) = ps(v.last)$ that capture the absolute pitch of the note n and $v.last$. Pitch values are taken from a pitch space in which C_4 has value 60 and a difference of 1 corresponds to one half step, e.g. C_5 has value 72. Using absolute pitches as separate input features will enable neurons on the hidden layer to discover possibly unknown pitch-based rules for perceptual streaming.

5.2.3 Temporal and Temporal Continuity Features

We define an inter-onset feature $\phi_4(n, v)$ as the temporal distance between the note onsets of n and $v.last$. An additional feature $\phi_5(n, v)$ is computed as the temporal distance between the note onset of n and the note offset (the time when a note ends) of $v.last$. These complementary features help our system model both acceptable rest lengths between notes and the gradual dissipation of note salience throughout the duration of a note.

Notes that lie between the onsets of $v.last$ and n may influence the voice assignment. Thus, we appropriately define a feature $\phi_6(n, v)$ as the number of unique onsets between the onsets of $v.last$ and n. We also define two features $\phi_7(n, v) = qd(n)$ and $\phi_8(n, v) = qd(v.last)$ for the durations of n and $v.last$, respectively, where note durations are measured relative to the quarter note. These features, when combined in the hidden layer, enable the system to learn to pair notes that appear in common duration patterns, such as dotted quarter followed by an eighth.

5.2.4 Chordal Features

Notes that reside in the soprano either alone or at the top of a chord tend to be heard as the most salient. As a result, the most prominent melodic line of a score often navigates through the topmost notes, even in situations where a candidate active voice lies closer in pitch to the alto or tenor notes of the current chord. Notes in a low bass range that stand alone or at the bottom of a chord exhibit a similar behavior. To enable the learning model to capture this perceptual effect, we define two features $\phi_9(n, v) = cp(n)$ and $\phi_{10}(n, v) = cp(v.last)$ to mark the relative positions of n and $v.last$ in their respective chords, where the chord position number (cp) starts at 0 from the top of a chord. To place chord positions into the appropriate context, we define $\phi_{11}(n, v)$ as the number of notes in n's chord and $\phi_{12}(n, v)$ as the number of notes in $v.last$'s chord. For more direct comparisons between notes in n's chord and the active voice, we calculate pitch proximities (*pd*) between $v.last$ and n's upper and lower neighbors $n.above$ and $n.below$. Thus, we define the features $\phi_{13}(n, v) = pd(v.last, n.above)$ and $\phi_{14}(n, v) = pd(v.last, n.below)$. We also add the features $\phi_{15}(n, v) = pd(n, n.above)$ and $\phi_{16}(n, v) = pd(n, n.below)$ to encode the intervals between n and its chordal neighbors.

5.2.5 Tonal Features

We use scale degrees $\phi_{17}(n, v) = sd(n)$ and $\phi_{18}(n, v) = sd(v.last)$ of the notes n and $v.last$ as features in order to enable the model to learn melodic intervals that are most appropriate in a given key. For example, if a candidate active voice ends on a leading tone, then it is likely to resolve to the tonic. We also define a feature $\phi_{19}(n, v)$ for the interval between the note n and the root of its chord, and similarly, a feature $\phi_{20}(n, v)$ for the interval between the note $v.last$ and the root of its chord.

The last tonal feature $\phi_{21}(n, v)$ is a Boolean feature that is set to 1 if the note $v.last$ in the active voice v appears in a tonic chord at a cadence. Tonic chords at cadences induce a sense of finality [1], which could potentially break the voice from the notes that follow.

5.2.6 Pseudo-polyphony Features

In pseudo-polyphony, two perceptually independent streams are heard within a rapidly alternating, monophonic sequence of notes separated by relatively large pitch intervals. Figure 9 presents an example of pseudo-polyphony.

Dataset	Model	All within-voice pairs of notes				Exclude pairs of notes separated by rests			
		Jaccard	Precision	Recall	F-measure	Jaccard	Precision	Recall	F-measure
Popular Music	Baseline	59.07	74.51	74.03	74.27	67.55	80.48	80.79	80.64
	NGModel	83.55	92.08	90.01	91.03	85.73	92.74	91.89	92.31
	ITA	92.45	94.96	97.21	96.08	93.04	95.12	97.70	96.41
Bach Chorales	Baseline	87.25	93.34	93.04	93.18	87.62	93.22	93.58	93.39
	NGModel	91.36	95.59	95.37	95.47	91.66	95.91	95.39	95.64

Table 2. Comparative results of Neural Greedy (NG) Model vs. Baseline on Popular Music and Bach Chorales; Inter-Annotator (ITA) results on the subset of 10 popular songs shown in Table 1.

Although the offset of each D_4 note is immediately followed by the onset of the next note, the often large intervals and the fast tempo break the upper and lower notes into two perceptually independent streams.

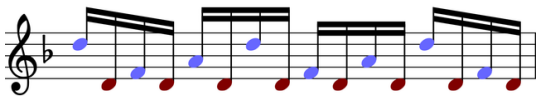

Figure 9. Example pseudo-polyphony from "Forest".

We model this phenomenon by introducing three features to the neural system. In designing these features, we first employ the envelope extraction method described in Section 5.1 to gather monophonic sequences of non-overlapping notes. We next find the maximal contiguous subsequences with an alternating up-down pattern of direction changes, like the one shown in Figure 9. The first feature $\phi_{22}(n, v) = apv(n)$ is set to be the alternating path value (*apv*) of the note n, which is 0 if n is not on an alternating path, 1 if it is in the lower part of an alternating path, and 2 if it is in the upper part of an alternating path. Similarly, we define $\phi_{23}(n, v) = apv(v.last)$ to be the alternating path value of the note $v.last$. The third feature is set to 1 if both n and $v.last$ have the same alternating path value, i.e. $\phi_{24}(n, v) = 1[apv(n) = apv(v.last)]$.

6. EXPERIMENTAL EVALUATION

We implemented the neural greedy model as a neural network with one hidden layer, an input layer consisting of the feature vector $\Phi(n, v)$, and an output sigmoid unit that computes the assignment probability $p(n, v|\theta)$. The network was trained to optimize a regularized version of the likelihood objective shown in Equation 1 using gradient descent and backpropagation. The model was trained and tested using 10-fold cross-validation. For evaluation, we considered pairs of consecutive notes from the voices extracted by the system and compared them with pairs of consecutive notes from the manually annotated voices. Table 2 shows results on the two datasets in terms of the Jaccard similarity between the system pairs and the true pairs, precision, recall, and micro-averaged F-measure. Precision and recall are equivalent to the soundness and completeness measures used in [6, 11]. We also report results for which pairs of notes separated by rests are ignored.

The results show that the newly proposed neural model performs significantly better than the envelope baseline, especially on popular music. When pairs of notes separated by rests are excluded from evaluation, the baseline performance increases considerably, likely due to the exclusion of pseudo-polyphonic passages.

Dataset	Model	Precision	Recall	F-measure
10 Fugues	[6]	94.07	93.42	93.74
	NGModel	95.56	92.24	93.87
30 Inv. 48 F.	[14]	95.94	70.11	81.01
	NGModel	95.91	93.83	94.87

Table 3. Comparative results on Bach datasets.

Close to our model is the data-driven approach from [6] for voice separation in lute tablature. Whereas we adopt a ranking approach and use as input both the note and the candidate active voice, [6] use only the note as input and associate voices with the output nodes. Therefore, while our ranking approach can label music with a variable number of voices, the classification model from [6] can extract only a fixed number of voices. Table 3 shows that our neural ranking model, although not specifically designed for music with a fixed number of voices, performs competitively with [6] when evaluated on the same datasets of 10 Fugues by Bach. We also compare the neural ranking model with the the approach from [14] on a different dataset containing 30 inventions and 48 fugues [1] .

7. CONCLUSION AND FUTURE WORK

We presented a neural model for voice separation in symbolic music that assigns notes to active voices using a greedy ranking approach. The neural network is trained on a manually annotated dataset, using a perceptually-informed definition of voice that also conforms to the musicological notion of voice as a monophonic sequence of notes. When used with a rich set of note-voice features, the neural greedy model outperforms a newly introduced strong baseline using iterative envelope extraction. In future work we plan to evaluate the model in the more general multi-label classification setting that allows notes to belong to multiple voices.

We would like to thank the anonymous reviewers for their helpful remarks and Mohamed Behairy for insightful discussions on music cognition.

[1] In [14] it is stated that soundness and completeness "as suggested by Kirlin [11]" were used for evaluation; however, the textual definitions given in [14] are not consistent with [11]. As was done in [6], for lack of an answer to this inconsistency, we present the metrics exactly as in [14].

8. REFERENCES

[1] E. Aldwell, C. Schachter, and A. Cadwallader. *Harmony and Voice Leading*. Schirmer, 4 edition, 2011.

[2] A. S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*. The MIT Press, Cambridge, MA, 1990.

[3] A. S. Bregman and J. Campbell. Primary Auditory Stream Segregation and Perception of Order in Rapid Sequences of Tones. *Journal of Experimental Psychology*, 89(2):244–249, 1971.

[4] E. Cambouropoulos. 'Voice' Separation: Theoretical, Perceptual, and Computational Perspectives. In *Proceedings of the* 9^{th} *International Conference on Music Perception and Cognition*, pages 987–997, Bologna, Italy, 2006.

[5] E. Chew and X. Wu. Separating Voices in Polyphonic Music: A Contig Mapping Approach. In *Computer Music Modeling and Retrieval:* 2^{nd} *International Symposium*, pages 1–20, 2004.

[6] R. de Valk, T. Weyde, and E. Benetos. A Machine Learning Approach to Voice Separation in Lute Tablature. In *Proceedings of the* 14^{th} *International Society for Music Information Retrieval Conference*, pages 555–560, Curitiba, Brazil, 2013.

[7] D. Huron. Tone and Voice: A Derivation of the Rules of Voice-Leading from Perceptual Principles. *Music Perception*, 19(1):1–64, 2001.

[8] A. Jordanous. Voice Separation in Polyphonic Music: A Data-Driven Approach. In *Proceedings of the International Computer Music Conference*, Belfast, Ireland, 2008.

[9] I. Karydis, A. Nanopoulos, A. N. Papadopoulos, and E. Cambouropoulos. VISA: The Voice Integration/Segregation Algorithm. In *Proceedings of the* 8^{th} *International Society for Music Information Retrieval Conference*, pages 445–448, Vienna, Austria, 2007.

[10] J. Kilian and H. Hoos. Voice Separation: A Local Optimization Approach. In *Proceedings of the* 3^{rd} *International Society for Music Information Retrieval Conference*, pages 39–46, Paris, France, 2002.

[11] P. B. Kirlin and P. E. Utgoff. VoiSe: Learning to Segregate Voices in Explicit and Implicit Polyphony. In *Proceedings of the* 6^{th} *International Society for Music Information Retrieval Conference*, pages 552–557, London, England, 2005.

[12] O. Lartillot. Discovering Musical Patterns Through Perceptive Heuristics. In *Proceedings of the* 4^{th} *International Society for Music Information Retrieval Conference*, pages 89–96, Washington D.C., USA, 2003.

[13] K. Lemstrom and J. Tarhio. Searching Monophonic Patterns within Polyphonic Sources. In *Proceedings of the* 6^{th} *Conference on Content-Based Multimedia Information Access*, pages 1261–1279, Paris, France, 2000.

[14] S. T. Madsen and G. Widmer. Separating Voices in MIDI. In *Proceedings of the* 7^{th} *International Society for Music Information Retrieval Conference*, pages 57–60, Victoria, Canada, 2006.

[15] D. Rafailidis, E. Cambouropoulos, and Y. Manolopoulos. Musical Voice Integration/Segregation: VISA Revisited. In *Proceedings of the* 6^{th} *Sound and Music Computing Conference*, pages 42–47, Porto, Portugal, 2009.

[16] D. Rafailidis, A. Nanopoulos, E. Cambouropoulos, and Y. Manolopoulos. Detection of Stream Segments in Symbolic Musical Data. In *Proceedings of the* 9^{th} *International Society for Music Information Retrieval Conference*, pages 83–88, Philadelphia, PA, 2008.

[17] D. Temperley. *The Cognition of Basic Musical Structures*. The MIT Press, Cambridge, MA, 2001.