1 Introduction

We propose a novel approach for automatically constructing new songs from a set of given compositions that involves sampling a melody line as well as the corresponding harmonies given by chords. We empirically show that songs generated by our approach are closer to music composed by humans than those of existing methods. Composing music is a complex creative process that is not only restricted by harmonic necessities but also by an implicit concept of good music. While music composition algorithms are able to comply with the explicit rules of composing, it is difficult to accord to the implicit concept of a good song. Moreover, this concept may vary between different music styles. We propose to learn the properties of good composed songs from a set of existing ones and use the learned model to construct new songs.

Furthermore, we use a hierarchical Markov model that captures transition probabilities between notes and chords as well as interdependencies between pitches, chords and durations. Additionally, our method uses a prior on the pitch distributions that takes into consideration the fact that most melodies can be clustered in sequences of up- or down-chains of notes. To allow for a song structure similar to modern pop songs, i.e., verses and choruses, the hierarchical model allows to predefined a block structure and then compose the songs such that blocks have a resembling melody.

The problem of automatic composing of songs has been tackled using machine learning techniques. Klinger and Rudolph (2007) generates the songs with a genetic algorithm that uses artificial neural networks to evaluate the fitness of each song. These fitness functions are trained using the feedback of human experts. Similarly, Fortier and Van Dyne (2011) employs genetic algorithms but the fitness is calculated based on score values assigned by human experts to predefined music features. In both approaches, no information is directly extracted from existing songs but the method is based on input from human experts. García Salas et al. (2011) use a probabilistic generative grammar similar to a Markov chain to generate a song. In this approach, the probabilities are learned from a training set of composed songs. Conklin and Witten (1995) predict the next note in the melody line by using a weighted combination of probabilities coming from multiple viewpoints. A viewpoint, or context model, is a higher order Markov chain which looks at only one particular feature of the previous notes, such as pitch, duration, musical interval from the pitch before or other similar note features. Similar to the previous approach, the context models are empirically estimated from a set of composed training songs. Common to all four approaches is that they only generate a melody line, whereas our approach is capable of generating a melody and an according harmony line of chords.

Two recent approaches are capable of generating melody as well as harmony. The music composing system APOCALEAPS (Sneyers and De Schreye 2010) generates songs using probabilistic rules and a Markov model for which the transition probabilities are manually set by human experts. The system Superwillow (Schulze and van der Merwe 2011) employs Markov models for which the transition probabilities are instead learned from a training set of composed songs. All mentioned methods have been evaluated using subjective feedback of humans. We also conduct a survey to evaluate our method but furthermore, we introduce a set of scores that measure the average similarity between generated songs and composed songs of a certain method. Theses scores are defined on
songs with a melody and a harmony line, so that we compare our approach directly with the two methods that are capable of generating melody as well as harmony.

2 STONES: Stochastic Technique for Generating Songs

Instead of using human experts to generate and evaluate songs, the method STONES constructs songs using a model learned on a set of examples. Each song consists of a melody line and a harmony line. The harmony line is a sequence of chords $c_1, \ldots, c_n$, i.e., a set of three or more harmonic notes taken from a predefined set of valid chords, with one chord per measure. The melody line is a sequence of notes, where each note is characterized by its pitch $p$ and duration $d$. Furthermore, in western music the melody line of a song consists of sequences of chains $s$ of notes, where each chain is a sequence of pitches going in the majority of cases in the same direction. Each chain is defined by its type $\tau \in \{\text{up, down, constant}\}$ and length $\lambda$ in number of measures. The sequence of chains is then used as a prior distribution on pitches to promote chain structures in songs but at the same time maintain variability.

We propose a Markov model that consists of three main parts: a chord model, a chain model and a melody model. The chord model is defined as

$$P(c_t = c) = \alpha_{c}P^{\text{prior}}(c) + \beta_{c}P^{\text{seq}}(c|c_{t-1}, \ldots, c_{t-k})\ ,$$

where $\alpha_{c}, \beta_{c} \in \mathbb{R}_{+}$ are weights with $\alpha_{c} + \beta_{c} = 1$. In this model, $P^{\text{prior}}(c)$ is the prior probability of chord $c$ appearing in a song and $P^{\text{seq}}(c|c_{t-1}, \ldots, c_{t-k})$ is the probability of chord $c$ appearing after chords $c_{t-1}, \ldots, c_{t-k}$. These probabilities are empirically estimated using a set of composed songs. Please note that, in accordance with musical practice, the model only considers the last $k$ chords to be relevant for the current one.

The chain model consists of a model for the chain type $\tau$ and one for the chain length $\lambda$. These models are defined as

$$P(\tau_t = \tau) = \alpha_{\tau}P^{\text{prior}}(\tau) + \beta_{\tau}P^{\text{seq}}(\tau|\tau_{t-1}, \ldots, \tau_{t-1})\ ,$$

$$P(\lambda_t = \lambda) = P(\lambda)\ ,$$

where $\alpha_{\tau}, \beta_{\tau} \in \mathbb{R}_{+}$ are again weights. The probabilities for the chain type are again empirically estimated using a training set of composed songs. For the chain length a Gaussian distribution is assumed, i.e., $P(\lambda) \sim \mathcal{N}(\mu, \sigma)$, where mean and variance are also estimated from the training set.

The melody model consists of a model for the pitch $p$ and one for the duration $d$ of notes in the melody. The duration is given as fractions of a 4/4-measure (in this paper we restrict all songs to a 4/4 measure, i.e., four beats per measure). The pitch model is defined as

$$P(p_t = p) = \alpha_{p}P^{\text{prior}}(p) + \beta_{p}P^{\text{seq}}(p|p_{t-1}, \ldots, p_{t-1}) + \gamma_{p}P^{\text{chord}}(p|c_0, \tau_t) + \delta_{p}P^{\text{pitch}}(p|p_{t-1})\ .$$

In this model, the chord $c_0$ is the one played simultaneously with the key $p_t$, i.e., $\theta_{c} = \lfloor \sum_{i=1}^{t} d_i \rfloor$. Similarly, the chain type $\tau_t$ is the type of the chain corresponding to pitch $p_t$, i.e., $\theta_{\tau} = \lfloor \sum_{i=1}^{t} \lambda_i \rfloor$.

The chain prior is defined as

$$P^{\text{chain}}(p|\tau_t, p_{t-1}) = \begin{cases} 1, & \text{if } p \text{ and } p_{t-1} \text{ respect direction of } \tau \\ 0, & \text{otherwise} \end{cases},$$

promoting pitches that adhere to the direction of the current chain. The duration model is defined as

$$P(d_t = d) = \alpha_{d}P^{\text{prior}}(d) + \beta_{d}P^{\text{seq}}(d|d_{t-1}, \ldots, d_{t-1}) + \gamma_{d}P^{\text{pitch}}(d|p_{t})\ .$$

The individual probabilities are again estimated using the training set. In algorithm 1 we provide the algorithm STONES$^{\text{chain}}$ for constructing a new song of length $L$ given these models.

In order to mimic the structure of typical pop songs, i.e., a sequence of similar blocks (e.g., verses and choruses) we propose a hierarchical extension of this algorithm, denoted STONES$^{\text{hier}}$. Given a predefined structure of a song in blocks $B_1, B_2, \ldots$, e.g., a verse-chorus-verse structure ($B_1, B_2, B_1$), the algorithm generates a separate song fragment for each block using STONES$^{\text{chain}}$. Aligning the song fragments according to the predefined structure results in an intermediate melody.
Algorithm 1 STONES\textsubscript{chain}

\textbf{Require:} Song length $L \in \mathbb{N}$, parameters $k, l, j \in \mathbb{N}$
weights $\alpha_c, \beta_c, \alpha_t, \beta_t, \alpha_p, \beta_p, \alpha_d, \beta_d, \gamma_p, \gamma_d \in \mathbb{R}^+$

\textbf{Return:} Song as sequence of chords $(c_1, ..., c_L)$ and notes $((k_1, d_1), ..., (k_m, d_m))$.

1: \textbf{for} $t=1$ \textbf{to} $L$ \textbf{do}
2: \hspace{1em} sample $c_t \sim P(c_t = c)$
3: \hspace{1em} $t \leftarrow 0$
4: \hspace{1em} \textbf{while} $\sum_{i=1}^{t} \lambda_i \leq L$ \textbf{do}
5: \hspace{2em} sample $\tau_t \sim P(\tau_t = \tau)$
6: \hspace{2em} sample $\lambda_t \sim P(\lambda_t = \lambda)$
7: \hspace{1em} $t \leftarrow t + 1$
8: \hspace{1em} $t \leftarrow 0$
9: \hspace{1em} \textbf{while} $\sum_{i=1}^{t} d_i \leq L$ \textbf{do}
10: \hspace{2em} sample $p_t \sim P(p_t = p)$
11: \hspace{2em} sample $d_t \sim P(d_t = d)$
12: \hspace{1em} $t \leftarrow t + 1$
13: \textbf{return} $(c_1, ..., c_L), ((k_1, d_1), ..., (k_m, d_m))$

The chords are directly adopted from the intermediate harmony. The final melody is then sampled according to the mixed distributions.

3 Experiments

For our experiments we use a collection of 85 songs from three different artists: The Beatles, Elton John and John Denver. We transpose all songs to the key of C major in order to have a common key. We use 65 of these songs to estimate the probabilities for our model: For both methods, STONES\textsubscript{chain} and STONES\textsubscript{hier}, we use the following parameters. The chord transition probabilities depend on the last $k = 4$ chords, the chain transition probabilities on the last $l = 2$ chains and the pitch and duration transitions on the last $j = 5$ notes. Figure 1 shows the score of the first few measures of a generated song.

![Generated song example](image)

Figure 1: Generated song example; The songs are generated in the key of C and then transposed to a random key (in this case, the key of D)

We evaluate the quality of songs generated by our method in two ways: a human survey and a score-based comparison with baselines. For the scoring we represent each song in a feature space of its empirical note, chord, and transition probabilities. The scores for a song depend on the distance of the song in this feature space in relation to a set of composed songs. The four scores are defined as follows.
We present the scores achieved by our methods in comparison with an uninformed baseline, i.e., songs where chord and note priors are uniform from the key of C, as well as the Superwillow and APOCALEAPS systems. We generate 14 songs with each of our methods as well as 14 with the uninformed baseline. Furthermore, we have obtained 14 songs generated by the APOCALEAPS system and 9 songs generated with the Superwillow system, both from the authors’ websites. The scores are calculated in relation to 20 composed songs from our collection that have not been used for model training. Table 1 presents the scores $s_1$ to $s_4$ for each method, averaged over the whole set of generated songs. The distance between songs is calculated in the previously mentioned feature space of note, chord and transition probabilities. In all scores, our methods outperform the uninformed baseline as well as the two music generation systems Superwillow and APOCALEAPS. For $s_1$ to $s_3$, the hierarchical method performs best, whereas in terms of score $s_4$ the simpler chain method is superior. In Fig. 2 we plotted all songs—generated and composed—in a 2D-embedding using the first two principal components of a PCA on the set of songs. The spectrum (also Fig. 2) indicates that already a few principal components have a high expressiveness. While the majority of songs generated by the uniformed baseline has a large distance to the cluster of composed songs, the ones generated by the other methods are not clearly visually distinguishable from the cluster of composed songs in 2D. In order to measure the closeness of generated songs to composed ones in the 2D embedding we calculated $s_1$ to $s_3$ also in the embedding (Table 1). For $s_1$ to $s_3$ our methods outperform all others significantly. For score $s_4$ the hierarchical method performs best. However, in this score the Superwillow systems performs better than the chain method. Overall, our methods outperform the baseline as well as the two music generation systems Superwillow and APOCALEAPS. We furthermore conducted a survey with 15 participants. For that, we generated 4 songs with STONES$_{\text{hier}}$, 4 songs with STONES$_{\text{chain}}$ and 4 uninformed songs. In the survey, we mixed the generated songs with 8 composed ones and asked the participants to classify the songs either as composed or generated. The survey confirms our hypothesis that songs generated by the hierarchical method are more likely to be confused for composed ones than those from the chain method which is again better than the uninformed baseline. A Wilcoxon signed-rank test on the collected data confirms the hypothesis with a significance level of $p \leq 0.01$. Particularly, in only 4 cases a participant misclassified an uninformed song as a composed song, while there were 22 cases in which a participant confused a hierarchical song with a composed one.

### 4 Conclusion

We presented a novel approach to constructing songs with melody, as well as harmony line from a set of given compositions. To evaluate our method we propose four scoring functions that measure
the similarity between generated songs and a set of composed ones. In terms of these scores, our method outperforms existing approaches. The structure of a song used in the hierarchical method is currently predefined. In future work we want to learn song structures from a collection of songs. Furthermore, we want to investigate to what extend the notion of chains can be generalized to more complex musical building blocks.

Acknowledgement

Part of this work was supported by the German Science Foundation (DFG) under the reference number ‘GA 1615/1-1’.

References


Jon Sneyers and Danny De Schreye. APOPCALEAPS: Automatic music generation with CHriSM. In G. Danoy et al., editors, *22nd Benelux Conference on Artificial Intelligence (BNAIC 2010)*, October 2010.