Musical Composer Identification

MUS-15

Niko Abeler

July 6, 2015

Introduction

Musical Composer Identification is a Music Information Retrieval task. Based on music sheets or recordings the composer of a musical piece has to be identified. The musical piece of question is unknown and the decision has to be based on previous work of the composer.

Most approaches are constructed for classical music and the amount of composers considered differs between 4 and 12. A variations of the problem is the decision between to composers, instead of of arbitrary many.

History

The Musical Composer Identification is commonly solved with machine learning techniques operating on features extracted from music sheets. A variety of techniques and features have been tested. The most successful approaches use Markov Models[1] or Neural Networks [2] and operate on discrete representations (MIDI or music sheets).

One of the first approaches to solve the Composer Identification was published by Pollastri and Simoncelli [3] in 2001. The described technique reached an accuracy of 42% but only considered 4 composers of classical music. Experts challenged with the same task reached an accuracy of 48%. Hidden Markov Models operating on relative pitch changes were created and trained for every composer.
Several approaches followed, working with a similar assumption; the author of a music piece can be identified by the note distribution. The presented techniques mainly differ in the choice of descriptors and machine learning structures used for evaluation. Common techniques were Naive Bayes and Support Vector Machines [4]. The descriptors were extracted from music sheets or discrete representations (e.g. MIDI, **kern).
Maximos A. Kaliakatsos-Papakostas et al. published two prominent approaches in 2010 and 2011. Currently dominant approaches rely on Hidden Markov Models or Neural Networks to evaluate the extracted features. The first approach[2] uses Feedforward Neural Networks (FNN) operating with Dodecaphonic Trace Vector measures the pitch distribution. The second paper[1] uses Weighted Markov Chain Models to distinguish between two composers. Both techniques use discrete representations and mainly focus on the pitch of notes.

Identifiers

The accuracy of an approach to solve the Musical Composer Identification heavily relies on the chosen identifier. Most identifiers are derived from music sheets and measure the note distribution of a song. An example for such identifiers is the Pitch Class Profile, which focuses on the pitch of notes while ignoring other properties.
Approaches relying on audio signals generally use identifiers derived from the frequency spectrum of the signal. A common example is Mel Frequency Cepstral Coefficients (MFCC), which are used for speech recognition but gets more popular in Music Information Retrieval.

Pitch Class Profile

The Pitch Class Profile (PCP) [2] is computed from music sheets or discrete representations of music and captures the commonness of notes. Notes are simplified to their pitch and normalized, meaning the there is no differentiation between notes in different octaves. PCP is a normalized version of the Chroma Profile.
The Chroma Profile $CP = (CP(1), CP(2), ..., CP(12))$ is 12-dimensional vector which counts the appearances of notes, in a musical piece M. $CP(n \mod (12) + 1)$ (1) is the summation of all notes at position n in the corresponding octave, e.g $CP(1)$ is the summation of all Cs.

$$CP(n \mod (12) + 1) = \sum_{n \mod (12) \in M}^{1} 1 \quad (1)$$

As musical pieces have different lengths, a normalization is applied(2), by dividing the CP by the most common note. The normalization is applied to receive a comparable representation.

$$PCP(i) = \frac{CP(i)}{\max_{1 \leq j \leq 12} CP(j)} \quad (2)$$

The Pitch Class Profile can be used as a global descriptor or as a local descriptor for a section of a song. When used as a global descriptor, the evolution of a musical piece is ignored and the song is reduced to its sole note distribution. More details are captured when using a fixed time windows approach. The musical piece is separated into windows of equal size and the PCP is calculated for each window. The resulting feature vector describes the evolution of the song. A sliding window approach captures even more information. A window with fixed size is slid above the music sheet and the PCP is calculated at each position.
PCP discards several properties that can be derived from music sheets. The pitch of the song is lost, as all notes are normalized to one octave. As the length of a note does not change its contribution to the PCP, the tempo and rhythm are lost. When used in a sliding window fashion the PCP preserves the melody.

Mel Frequency Cepstral Coefficients

The Mel Frequency Cepstral Coefficients (MFCC) were developed in 2000[5] and are commonly used in speech recognition tasks, but are getting more popular for musical information retrieval tasks. By integrating the mel scale, MFCC takes the human perception into account. MFCC is relatively robust, but behaves poorly in the presence of additive noise. MFCCs are commonly computed for short (3 seconds) audio fragments. The Fourier series of the signal is calculated and mapped to the log mel scale(3), which describes the human perception of a tone, based on its frequency.

$$m = \log(2595 \log_{10}(1 + \frac{f}{700})) \quad (3)$$

The MFCCs are obtained by forming the Discrete Cosine Transform of the mel log powers.
A variant of Mel Frequency Cepstral Coefficients are the Mel Phone Coefficients, where the the Discrete Cosine Transform is not applied.
...

Deep Neural Network

Zhen Hu et al.[6] proposed the usage of a Deep Neural Network to solve the Musical Composer Identification. Their approach describes a 5-layer Deep Neural Network (DNN) operating on 30 second long audio clips. Mel Phon Coefficients are extracted from the input signal and processed by Denoising Autoencoders. The refined features are evaluated by two Restricted Boltzmann Machines and mapped to a composer by a logistic regression.
Although the approach works with potentially noisy data, its accuracy is comparable to recent techniques working on music sheets.

Input

The Deep Neural Network operates on 30 second long audio clips. The input clip is separated into 3 seconds fragments with an overlay of 50%. For every fragment the MPC is computed, resulting 592 coefficients used in the next layer.

Denoising Autoencoder

As the DNN operates with recordings of different quality and properties, a denoising is necessary to create a robust algorithm. This is achieved by two Denoising Autoencoders. Denoising Autoencoders (DA) were proposed by Pascal Vincent et al. [7] in 2008. They inputs to intermediate representations for neural networks and are robust to partial corruption.

A Denoising Autoencoder maps the input x to a hidden reprenstation y with trained weights W and offsets b (5). The hidden representation is mapped to the output z (6). Mapping to hidden representation and output are done in similiar fashion. The input is weighted and an offset is applied and the sigmoid function is applied to every vector entry (4). The sigmoid function can be replaced by other non-linear functions.

$$sgm(x) = \frac{1}{1+e^{-x}}$$

$$sgm((x_0, ..., x_n)) = (sgm(x_0), ..., sgm(x_n)) \quad (4)$$

$$y(x) = sgm(Wx + b) \quad (5)$$

$$z(y) = sgm(W^T y + b') \quad (6)$$

The DA is trained to restore corrupted input vectors. For the training a corrupted version of the training set is created by setting some entries to zero. Weights and offset are chosen to minimize the error between the restored and original vectors (7).

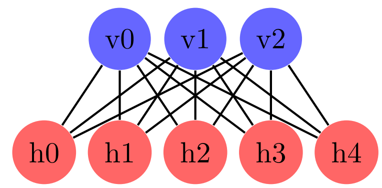

Figure 1: Graph describing a RBM with three visible and 5 hidden nodes

$$\min_W ||sigm(W^T \cdot sigm(W \cdot x')) - x||_2 \quad (7)$$

Restricted Boltzmann Machine

Restricted Boltzmann Machines (RBM) are an energy-based models. A RBM consist of hidden and visible nodes as depicted in 1. The energy of a RBM is described by a linear equation (8) with weights W and offsets b and c. The weights W are depicted by the edges in 1. Visible nodes serve as input nodes, while the hidden nodes contain the output.

$$E(v, h) = -h^T W v - b^T v - c^T h \quad (8)$$

In the training of a RBM weights and offsets are chosen to maximize the probability(9) of the training set V.

$$P(V) = \prod_{v \in V} p(v) \quad (9)$$

For given input $v_0, ..., v_n$ the hidden nodes $h_0, ..., h_m$ are chosen to minimize the probability $p(v, h)$ 10, where Z is a normalizing factor. The evaluation of the RBM is typically preformed by Gibbs Sampling[8].

$$p(v, h) = \frac{e^{-E(v,h)}}{Z} \quad (10)$$

Accuracy

The Deep Neural Network was trained with 250 clips 30 seconds. The clips were randomly picked from different recordings. 50 clips

Composer	Accuracy	Composer	Accuracy
Bach	93.10%	Haydn	40.00%
Beethoven	63.33%	Mendelssohn	100.00%
Brahms	75.51%	Mozart	74.58%
Chopin	98.11%	Schubert	20.59%
Dvorak	97.01%	Vivaldi	87.04%
Handel	100.00%		

Table 1: Accuracy for each considered composer

served as a validation set and the test set was composed of 60 clips. The choice for clips in the training set was restricted. If a clip of a musical piece was present in the training set, no clip of this piece could be present in the test set. This restriction avoids learning something about specific musical pieces or recordings.

The approach reaches an overall accuracy of 76.26%, but differs significant between composers, as shown in table 1. While the overall accuracy is high the approach particularly struggles with Haydn and Schubert. The paper gives no identification for the bad performance. It might be result of insufficient training data for the composers or a high similarity between Schubert, Haydn and other composers.

Other approaches observed similar behavior, where two composers showed a high similarity. [2] for example struggles with the differentiation between Haydn and Mozart. This could be an explanation for the poor performance of the Deep Neural Network for Haydn, as Mozart also lays below the average accuracy.

Conclusion

The Musical Composer Identification is still a hard problem, although the accuracy rises and an overall improvement and be observed. Recent approaches reach an accuracy between 60% and 80%, raising the need for further assessment of results of experts. Most techniques are operating on music sheets, which lays in the nature of the problem, as it is mostly about classical music.

References

[1] Maximos A Kaliakatsos-Papakostas, Michael G Epitropakis, and Michael N Vrahatis. Weighted markov chain model for musical composer identification. In *Applications of Evolutionary Computation*, pages 334–343. Springer, 2011.

[2] Maximos A. Kaliakatsos-papakostas, Michael G. Epitropakis, and Michael N. Vrahatis. Musical composer identification through probabilistic and feedforward neural networks, 2010.

[3] Emanuele Pollastri and Giuliano Simoncelli. Classification of melodies by composer with hidden markov models. In *Web Delivering of Music, 2001. Proceedings. First International Conference on*, pages 88–95. IEEE, 2001.

[4] David Yu Justin Lebar, Gary Chang. Classifying musical scores by composer: A machine learning approach, 2008.

[5] Beth Logan et al. Mel frequency cepstral coefficients for music modeling. In *ISMIR*, 2000.

[6] Zhen Hu, Kun Fu, and Changshui Zhang. Audio artist identification by deep neural network. *CoRR*, abs/1301.3195, 2013.

[7] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[8] George Casella and Edward I George. Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.