Content-based Music Recommendation

MUS-17

Kinan Halloum

July 5, 2017

Abstract

Many music recommendation algorithms are based on the idea of collaborative filtering, in which new music is recommended to users based on their listening behavior only e.g. track playcounts and user similarity. However, these approaches do not work well for new, unpopular music tracks, as only little usage data is available. This is known as the "cold start problem", and in this seminar, a deep learning based solution, described in [4], is presented.

Papers of interest

I chose van den Oord et al. [4] to be the main paper on which the presentation is based, as it's one of the very few -if not the only to date- papers to apply deep learning on audio data directly for the purpose of recommending music, which I found to be very interesting. In Hu et al. [1], the principles of WMF are described, and are included in [4] as part of their approach.

In addition, one of the authors of [4] was interning at Spotify, which is a major music content provider and I use their music client on daily basis.

Collaborative filtering

Collaborative filtering (CF) is a widely used method for recommending items to consumers, mainly on commercial platforms such as music/video streaming services and online shops.

Model-based collaborative filtering

Given a set of tracks (items) and one of users (consumers), CF can be applied by finding real-valued vectors U, I called latent vectors, for each user u and track i respectively, such that their dot product is relatively large if and only if the probablity of user u liking i is high and vice versa. This is achieved by using an optimization algorithm e.g. Weighted Matrix Factorization (WMF) based on the available usage data, mainly being track playcounts.

Weighted Matrix Factorization

The WMF algorithm takes a so called "confidence" matrix $\mathbf{C}$ as its input and outputs two matrices $\mathbf{U}$, $\mathbf{I}$ with $\mathbf{U} \cdot \mathbf{I} \sim \mathbf{C}$, where $\mathbf{U}$, $\mathbf{I}$ contain the latent vectors of users resp. tracks. The initial confidence matrix $\mathbf{C}$ is calculated from user(u)-track(i) playcounts r_{ui} according to the following formula [4]:

$$c_{ui} = 1 + \alpha \log(1 + \epsilon^{-1} r_{ui})$$

where α, ϵ are real-valued parameters. The matrix factorization is the calculated by optimizing the following objective function [4]:

$$\min_{U_*, I_*} \sum_{u,i} c_{ui}(p_{ui} - U_u^T I_i)^2 + \text{Regularization term}$$

where p_{ui} is called the preference for user u and track i and is equal to one if $r_{ui} > 0$ and zero otherwise. U_u, I_i are the latent vectors for user u, track i respectively.

The cold start problem

Because CF is purely based on users' usage data, recommending unpopular music tracks is problematic. Technically speaking, their corresponding latent vectors would be under-constrained and their values would largely depend on the initialization which is usually random. As unpopular music tracks form the majority of the total tracks there is, it is important to find solutions for the cold start problem.

Content-based MR

One way of approaching the cold start problem is by incorporating the track's audio data in the recommendation process. However, it has been found that there is no direct correspondence between the raw audio signal and the characteristics which affect user preferences, which is known as the **semantic gap problem**. This makes extracting appropriate features for the purpose of MR a rather challenging task. Although there are algorithms which extract specific track features e.g. Genre and mood, relying on a limited set of features might not be well suited for the task of MR as liking a track is much more complicated than just having a favorite Genre.

(Deep) Neural networks

Neural networks (NNs) have the potential to solve the above problem by automating the task of feature extraction.

NNs have gained lots of attention during the past years. One main reason for that is the rapid increase in computational capacity of computers, in addition to the advancements in understanding the way NNs work.

The main function of NNs is to recognize/differentiate certain characteristic, i.e. features in the input data without having to fully memorize it. Nowadays, NNs are used in many applications such as pedestrian tracking, image segmentation, text recognition etc. However, until recently, little has been done in the field of audio as opposed to that of vision. One major reason for this is the fact that most audio files such as music tracks, have a relatively long one dimensional signal. For example, a three-minute long track with a typical sampling rate of 44100 Hz would result in $3 \cdot 60 \cdot 44100 = 7938000$ values, which amounts to almost three times the number of values for an image of size 1920 by 1080 pixels, although the typical image size in common training sets is much smaller.

Convolutional NNs (CNNs) If the features one would like the NN to differentiate are local, then there exists a special type of NNs, called convolutional CNNs, where calculation are much more efficient, which allows to work on large data units, such as images and audio files, as features in this case are mostly local, e.g. the beginning of a music track has not much to do with its end.

CNNs are used as a part of the algorithm described in [4].

Related work

There have been some attempts to overcome the cold start problem by making use of content-based features, e.g. in [5] for the purpose of scientific article recommendation or in [6] for collaborative music retrieval (query-based recommendation). However, non of which have used deep neural networks.

Main approach

The main idea behind the method explained in [4] is to first compute the latent item(track)- and user-factors. This is done by applying the WMF algorithm to a large dataset[1] that includes the playcounts of over 380,000 songs, collected from 1 million users.

The obtained latent track-factors are then used as ground truth for training a CNN along with the corresponding audio data as its in-

[1] The Echo Nest Taste Profile Subset

put. To this regard, the audio signals from each track are first converted into a fixed-size **mel-spectrogram**. The spectrograms where generated from random 3 second snippets of audio, this was mainly done to improve the CNN's training performance.

Finally, the trained neural network is used to predict the latent item-factors for tracks with little to no usage statistics by solely relying on its audio data (content-based).

A personal note on mel-spectrograms

Mel-spectrograms are a method of "compressing" the wave form audio into a representation that is closer to the way humans perceive sound. One might ask why haven't the author of [4] used the wave form directly as an input to the network. In my opinion, converting audio to spectrograms can be seen as more of a way of compressing audio, than of extracting certain features. This is comparable to using the JPEG representation of images instead of the pixel values. In addition, using the wave form is rather more computationally demanding than using the corresponding mel-spectrogram, that is why they are widely used for many audio-classification tasks. However, recently there have been a tendency towards using the wave form directly as an input to the NNs, e.g. [2, 3]

Loss functions

The loss function is a necessary component of NNs, which plays a major role in what and how fast a neural network learns specific features of the input. It is a measure of how well the current NN does reproduce the target values -in our case the track latent vectors-, and this has a direct influence on the "learning" behavior of the network as training it is guided by trying to minimize its loss function. A commonly used loss function is the **mean squared error (MSE)**, which is the average of the squared differences of the network's output and the training target value.

In [4], two networks were trained using two different objective functions:

$$(1)\ \min_{\theta} \sum_{i} ||I_i - I_i^{'}||^2$$

$$(2)\ \min_{\theta} \sum_{u,i} c_{ui}(p_{ui} - U_u^T I_i)^2$$

The function (1) is based on the MSE, where the function (2) is based on the WMF objective function.

Results

The following table is some of the interesting quantitative results obtained by [4]

	Mean Average Precision
random	0.00015
CNN with (1) as loss	0.00672
CNN with (2) as loss	0.23278

It is clear from the above table that using the WMF objective function to train the network is much better than using the MSE, however, it is important to note, that when using (2) as a loss function, an upper bound of the mean average accuracy was calculated. Practical values were often lower.

The authors have also mentioned that their quantitative results do not properly reflect their qualitative ones, which were better, which was due to the way the mean average precision was calculated, as it has considered audio meta data e.g. year, artist, which cannot be predicted from raw audio.

My contribution

Placeholder

Conclusion

In this seminar, an approach, described in [4], to overcome the cold start problem was explained. The main idea included training a CNN on the track's latent vectors as target values and its audio wave form as its input. In conclusion, results showed that using this approach for the purpose of music recommendation is relatively better than randomly suggesting tracks.

References

[1] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, Dec 2008.

[2] Jongpil Lee, Jiyoung Park, Keunhyoung Luke Kim, and Juhan Nam. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. *arXiv preprint arXiv:1703.01789*, 2017.

[3] Shuhui Qu, Juncheng Li, Wei Dai, and Samarjit Das. Understanding audio pattern using convolutional neural network from raw waveforms. *CoRR*, abs/1611.09524, 2016.

[4] Aaron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems 26*, pages 2643–2651. Curran Associates, Inc., 2013.

[5] Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 448–456, New York, NY, USA, 2011. ACM.

[6] Jason Weston, Chong Wang, Ron Weiss, and Adam Berenzeig. Latent collaborative retrieval.