DEEP RANKING: TRIPLET MATCHNET FOR MUSIC METRIC LEARNING

Rui Lu[1] *Kailun Wu*[1] *Zhiyao Duan*[2*] *Changshui Zhang*[1†]

[1] Department of Automation, Tsinghua University
State Key Lab of Intelligent Technologies and Systems
Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, P.R.China
[2]Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY USA

ABSTRACT

Metric learning for music is an important problem for many music information retrieval (MIR) applications such as music generation, analysis, retrieval, classification and recommendation. Traditional music metrics are mostly defined on linear transformations of handcrafted audio features, and may be improper in many situations given the large variety of music styles and instrumentations. In this paper, we propose a deep neural network named Triplet MatchNet to learn metrics directly from raw audio signals of triplets of music excerpts with human-annotated relative similarity in a supervised fashion. It has the advantage of learning highly nonlinear feature representations and metrics in this end-to-end architecture. Experiments on a widely used music similarity measure dataset show that our method significantly outperforms three state-of-the-art music metric learning methods. Experiments also show that the learned features better preserve the partial orders of the relative similarity than handcrafted features.

Index Terms— Metric learning, music similarity, deep learning, convolutional neural networks

1. INTRODUCTION

Automatically learning metrics to measure music similarity is an important problem in Music Information Retrieval (MIR) with many applications including music recommendation, classification, and search. Compared to data-independent metrics such as the Euclidean distance, learned metrics often better capture structures in the data and suit the tasks at hand [1, 2, 3, 4, 5, 6]. Metric learning is most often conducted in a supervised fashion: the learning algorithm is trained on examples with human-annotated ground-truth similarity ratings. Unsupervised metric learning approaches, such as Mahalanobis distance, Principal Component Analysis (PCA) and other dimensionality reduction algorithms, usually do not achieve as good performance as supervised approaches [7].

*ZD acknowledges the National Science Foundation grant no. 1617107.
†CZ acknowledges the funding by 973 Program (2013CB329503), NSFC (Grant No.61621136008, and No.61473167) and the German Research Foundation (DFG) in Project Crossmodal Learning DFC TRR-169.

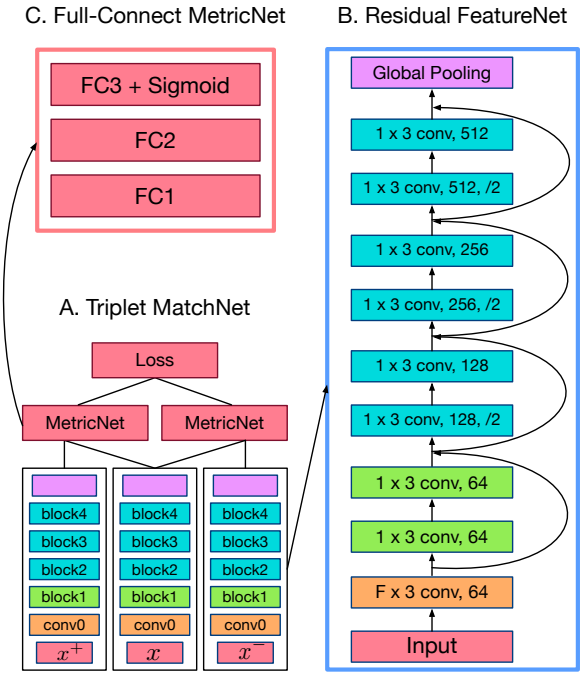

Fig. 1. Structure of the proposed Triplet MatchNet.

The annotation of ground-truth similarities between music pairs is difficult and time consuming, as one has to make sure the annotation criterion is consistent over a large number of pairs [8]. Therefore, some methods consider to learn and predict *relative similarity* [1, 2, 3, 4] instead. One type of relative similarity is to consider triplets of examples in a dataset $\mathbb{X}$. Specifically, one example in a triplet is a query χ and the other two are ranked as a more similar example χ^+ and a less similar example χ^- to the query. Then metric learning can be formulated as the learning of a distance embedding function $f : \mathbb{X} \times \mathbb{X} \longmapsto \mathbb{R}^+$ that maps the query closer to the more similar example than to the less similar one:

$$f(\chi, \chi^+) < f(\chi, \chi^-), \forall \chi \in \mathbb{X}. \quad (1)$$

For music data, similarity annotations are often at song level. Metric learning algorithms, however, should perform at

Residual FeatureNet: Convolution and Global Pooling			
layer	**out-size**	**filters**	**shortcut**
conv0	$1\times T$	$F\times 3$, 64, (1, 1)	-
block1	$1\times T$	1×3, 64, (1, 1) 1×3, 64, (1, 1)	Identity
block2	$1\times \lceil T/2 \rceil$	1×3, 128, (1, 2) 1×3, 128, (1, 1)	1×1, 128, (1, 2)
block3	$1\times \lceil T/4 \rceil$	1×3, 256, (1, 2) 1×3, 256, (1, 1)	1×1, 256, (1, 2)
block4	$1\times \lceil T/8 \rceil$	1×3, 512, (1, 2) 1×3, 512, (1, 1)	1×1, 512, (1, 2)
GP	1×512	-	-
Full-Connect MetricNet			
layer	**out-size**	**filters**	**shortcut**
FC1	W	-	-
FC2	W	-	-
FC3	1	-	-

Table 1. Model details: inputs are frames with size $C\times F\times T$ (channels' number$\times$ frequency bands' number $\times$ time hops' number); filters are denoted as "(frequency bands' number $\times$ time hops' number), filters' number, (frequency stride, time stride)"; all activation functions are omitted for brevity; GP means global pooling; W is determined through experiments.

a much smaller time scale and then achieve song-level similarity through post-processing for two reasons: first, in applications such as music recommendation, similarity is defined along short-term aspects such as timbre and harmony; second, long-term modeling in music is very challenging. Therefore, one way to balance the above contradiction is to learn frame-level music metric and further obtain song-level similarity by majority vote. We utilize this strategy and detail it in Sect.2.

To the extent of our knowledge, no nonlinear metric learning methods have been applied to MIR. Methods dealing with relative similarity for music typically learn linear projections such that distances between similar pairs are minimized while those between dissimilar pairs are maximized. Typical relative metric learning methods for MIR include relative information theoretic metric learning (RITML) [1], metric learning to rank (MLR) [3] and an SVM-based approach [4]. These methods operate on handcrafted song-level features, which are simple statistics of frame-level features. In addition, linear projections are unlikely to capture complex patterns in data that often only emerge after nonlinear transformations.

Deep learning methods are promising in learning nonlinear features from raw data in various domains such as computer vision [9, 10, 11], natural language processing [12] and MIR tasks [13, 14, 15, 16, 17]. Regarding metric learning, Han *et al.*[9] proposed a deep network called MatchNet that learns similarities between similar and dissimilar image pairs. To our best knowledge, no deep models have been proposed for rank-based metric learning nor for music applications.

In this paper, we propose a deep model called *Triplet MatchNet* by extending the rank-based metric learning [3] to end-to-end training framework and take advantages of deep models that automatically learn nonlinear features from raw data. As in Fig.1A, Triplet MatchNet comprises of a triplet of residual nets [18] at bottom for feature extraction and two fully connected nets on top to obtain distances between query χ and the similar/dissimilar examples χ^+/χ^-. Our model is closely related to MatchNet [9]. However, MatchNet is classification-based and models absolute similarity, while Triplet MatchNet is rank-based and learns from relative similarity of audio triplets. Moreover, relative similarity is much easier to collect for music data and the proposed method is preferred. We conduct experiments on the MagnaTagATune music similarity dataset [19] and compare our approach with three state-of-the-art methods. Results show that the proposed method surpasses the comparison methods significantly on the constraints fulfillment task. Further analyses show that features learned by the proposed method (outputs of the residual net) can better preserve distance constraints.

2. METHOD

As described in Eqn.(1), our goal is to learn a distance embedding function f from triplets of songs $< \chi, \chi^+, \chi^- >$. In order to model the rich variations of audio signals of the music, we propose to first learn the embedding at frame-level, and then calculate song-level relative similarity by majority vote. Let us denote the frame set of song χ as $\{x\}$, and those of songs χ^+/χ^- as $\{x^+\}/\{x^-\}$ respectively. Then the distance embedding function learned by Triplet MatchNet is:

$$f(x, x^+) < f(x, x^-) \quad \forall x \in \chi, x^+ \in \chi^+, x^- \in \chi^-. \quad (2)$$

The above formulation tries to learn an f that maps all frames of the query song χ closer to any frame of the more similar song χ^+ than to any frame of the less similar song χ^-.

2.1. Data Preprocessing

We down sample all songs to 16 kHz, apply Hann window and STFT with three window lengths (1024, 2048, and 4096 points) and the same hop size of 512 to get three magnitude spectrograms. Next, we apply 80-band mel-scale triangular filter ranging from 0 Hz to 8 kHz to obtain mel-scale spectrograms. We normalize each frequency band to zero mean, unit variance and stack the three spectrograms to make a 3-D tensor. Finally, we aggregate every 50 (about 1.824 seconds) adjacent windows without overlap to form frames. Note that the length of a bar in pop songs is around 2 seconds, frames generated above are capable of capturing meaningful timbre and harmony changes. To sum up, each song is divided into a set of frames that form inputs in Fig.1 and each frame is a $3 \times 80 \times 50$ tensor. Thus in Table 1, $C = 3, F = 80, T = 50$.

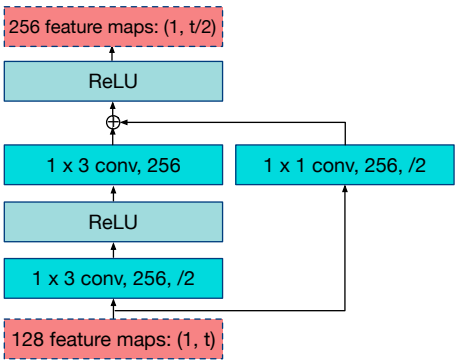

Fig. 2. Structure of the residual block

2.2. Network Architectures

Triplet MatchNet is a deep network (Fig.1A) which simultaneously learns a residual feature net (Fig.1B) and a fully connected metric net (Fig.1C). Details are shown in Table 1.

We utilize the residual networks [18] for feature extraction (Figure1B) due to its performance gains from the relieving of gradient vanishing problem by shortcut connections. A set of 64 convolutional filters with size (F, 3) act as the beginning convolutional layer, resulting in 64 feature maps with size (1, T). These filters span the entire frequency range to allow no shift invariance along frequency axis. We then add 4 consecutive temporal residual blocks with 1×3 filters (we only apply transformations on temporal domain afterwards) following design rules by ResNet [18] and VGG [20]. We perform downsampling by convolutional layers with strides of 2. Feature extraction stage ends with a global mean-pooling. The three residual networks (Fig.1B) share the same parameters; updates for either net will be applied to all the nets.

As shown in Fig.1B, the main part of the residual network contains 4 blocks, each a stack of two convolutional layers and the kernel sizes of all the temporal convolutions are 1×3 with zero paddings when necessary such that the temporal resolution is conserved or halved when the convolutional operation is carried out with stride 2. We take block3 as an example and detail its structure in Fig.2. Rectangles with dashed borders are input and output feature maps.

Our metric networks (Fig.1C) are inspired by recently proposed methods that make use of fully connected layers to calculate distances between extracted features [9, 10, 11]. It comprises of two fully connected layers with ReLU nonlinearity and another fully connected layer with sigmoid as the final output. Same as the residual networks, the two metric nets share parameters and we update them at the same time.

2.3. Network Training

For a triplet of songs $< \chi, \chi^+, \chi^- >$ described in Eqn.(1), we represent them as frame sets $\{x\}$, $\{x^+\}$ and $\{x^-\}$ as described in Sect.2.1. The two distances output by the Triplet MatchNet can then be represented as:

$$d^+ = f(x, x^+) = h(g(x), g(x^+)),$$
$$d^- = f(x, x^-) = h(g(x), g(x^-)), \quad (3)$$

where $g : \mathbb{R}^{C\times F\times T} \mapsto \mathbb{R}^{512}$ and $h : \mathbb{R}^{512} \times \mathbb{R}^{512} \mapsto [0, 1]$ denote operations by residual networks and metric networks.

In traditional rank-based metric learning [3], the undifferentiable *partial order* is used as an objective to be maximized. We modify it to be a continuous loss suitable for our model:

$$\psi(x) = \frac{1}{|\{x^-\}|} \sum_{x^-\in\{x^-\}} \max\{0, d^+_{max} - f(x, x^-)\}, \quad (4)$$

where $d^+_{max} = \max_{x^+\in\{x^+\}} f(x, x^+)$ is the maximum value among all the distances from x to frames in the more similar song. Besides the description of partial order in Eqn.(4), we force distances between query frame x and the more similar song $\{x^+\}$ to 0 while distances between x and $\{x^-\}$ to 1:

$$\phi(x) = -\frac{\sum_{x^+}\sum_{x^-}[\log(1 - d^+) + \log(d^-)]}{|\{x^+\}||\{x^-\}|}. \quad (5)$$

Eqn.(5) maximizes the manifold margin between positive and negative frames, which in turn benefits the conservation of the partial order. Thus, the final loss for a triplet of songs fed into Triplet MatchNet for training as shown in Fig.1A is:

$$loss(\chi, \chi^+, \chi^-) = \frac{1}{|\{x\}|} \sum_{x\in\{x\}} (\psi(x) + \phi(x)). \quad (6)$$

Given the trained model, we calculate d^+/d^- in Eqn.(3) for any frame triplet (x, x^+, x^-) in song triplet and $d^+ < d^-$ means that the frame-level constraint is fulfilled. We can further determine whether the song-level constraint is reserved by majority voting through considering all the frame triplets.

3. EXPERIMENTS

3.1. Setup

We evaluate the proposed method on the widely used MagnaTagATune[1] dataset. It contains both audio recordings and song-level relative similarity annotations in the form as "$\chi+$ is more similar to χ than χ^-". Following the preprocessings in [8], we obtain a total of 860 triplet constraints (χ, χ^+, χ^-) involving 993 unique songs, each 29-second long. We evaluate our model and the baselines using the constraints fulfillment rate (CFR) [1, 2, 3, 4], which is defined as the percentage of song-level constraints satisfied by the learned distance function. For the proposed method, we use the Adam [21] optimization algorithm and dropout with $p = 0.5$ on the first two fully connected layers of our metric network to avoid overfitting. The dimension of the first two dense networks, W, is set to 512, as we find it provides the best performance. We compare the proposed method with five baselines: RITML [1], RMLR [2], MLR [3], SVM [4] and the Euclidean distance.

[1] http://mirg.city.ac.uk/codeapps/the-magnatagatune-dataset

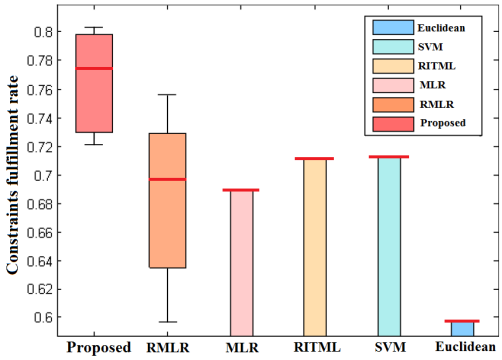

Fig. 3. Constraints Fulfillment Rate (CFR) comparison with 10-fold cross validation. Triplet MatchNet and RMLR are run by ourselves hence are shown as boxplots. Features input to RMLR are 512-d PCA-reduced mel-features. The other baselines are shown as barplots, whose values are taken from [1], which reports average CFR through 10-fold cross validation on the same dataset with a likely different partition.

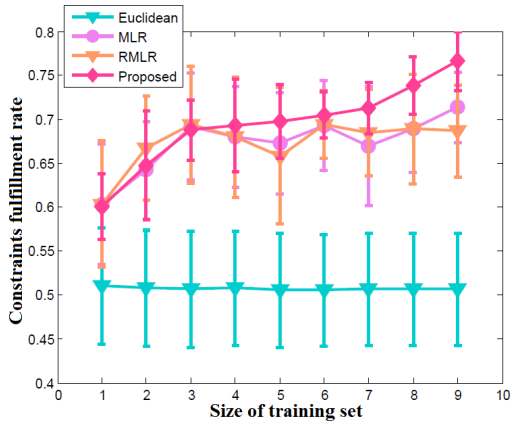

Fig. 4. Generalization capability with 10-fold cross validation when the training set size changes. The horizontal axis indicates the number of folds included in the training set. Mean (dots) and standard deviation (bars) are reported.

3.2. Results and Discussions

For the first experiment, we report the overall CFR comparison among the proposed method and the five baselines through 10-fold cross validation following [1, 8]. Since the baseline methods differ from the proposed method in both the model and features and we do not have access to their features, we cannot run these baselines in their original settings. Fortunately, four baselines (MLR, RITML, SVM, and Euclidean) have been compared and reported with their average CFR through 10-fold cross validation on the same dataset in [1]. Therefore, we simply report these average CFR values as barplots in Fig. 3. For the RMLR baseline, we run it by ourselves and generate a boxplot as the proposed method. We compress the $3 \times 80 \times 50$ 3D mel-spectrograms via PCA to maintain 95% of the variance as the feature inputs for RMLR following [2]. We empirically set $C = 10^{-2}, \lambda = 10^{-3}$ and fix Δ to AUC for training. For further details of choosing Δ, please refer to [3]. We can see that our model outperforms all baselines by a large margin. Moreover, it is necessary to note that [1] combines a variety of features such as chroma, timbre and tempo to form song-level features while our model automatically learns representations from raw mel-scale spectrograms. This suggests that the features learned by the highly nonlinear deep networks may better capture the music similarity than the handcrafted features.

Method	HandCrafted	PCA	Residual
RMLR	-	65.9 ± 8.3	$\mathbf{71.2 \pm 7.2}$
MLR	68.9	61.7 ± 10.5	$\mathbf{71.7 \pm 6.9}$
Euclidean	59.8	50.7 ± 6.3	$\mathbf{70.6 \pm 3.8}$

Table 2. Constraints Fulfillment Rate (CFR) of three baselines working with different features: the original handcrafted features as reported in [1], the vectorized mel-features reduced to 512-d by PCA, and features learned by residual networks of the proposed approach.

The second experiment evaluates how these methods generalize when the size of training data changes as in Fig. 4. Again, we randomly split the dataset into 10 folds. We run the experiment 10 times, each time we test the CFR on one fold and gradually enlarge the training set, which is composed by a subset of the other folds. By gradually enlarging the training set, performances of our method and the baselines become better in general. The proposed method shows better performance over the baselines across almost all sizes of the training set except for those cases when training sets are too small.

The third experiment exhibits effectiveness of the features extracted by the proposed method. We use it as a feature extractor and combine it with three baselines (RMLR, MLR, and Euclidean). We report the average CFR of a 10-fold cross validation in Table 2. We compare three settings: 1) the results reported in [1] where their original handcrafted features are used, 2) the PCA-reduced mel-features as described in the first experiment, and 3) the features extracted by the residual networks in the proposed method. The results show that features learned by our residual networks better preserve partial orders and boost performance of the traditional algorithms.

4. CONCLUSIONS

We proposed a deep structure to learn nonlinear metrics from relative similarity annotations of song triplets. Thanks to the powerful capability of deep representations of the residual network and the complex nonlinearities of the fully connected metric network, we achieved significantly better performance than the traditional music metric learning methods.

5. REFERENCES

[1] Daniel Wolff, Andrew MacFarlane, and Tillman Weyde, "Comparative music similarity modelling using transfer learning across user groups," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[2] Daryl KH Lim, Brian McFee, and Gert Lanckriet, "Robust structural metric learning," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2013.

[3] Brian McFee and Gert Lanckriet, "Metric learning to rank," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.

[4] Matthew Schultz and Thorsten Joachims, "Learning a distance metric from relative comparisons," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2003.

[5] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon, "Information-theoretic metric learning," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2007.

[6] Kilian Q Weinberger and Lawrence K Saul, "Distance metric learning for large margin nearest neighbor classification," *The Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2009.

[7] Brian Kulis, "Metric learning: A survey," *Foundations and Trends in Machine Learning*, vol. 5, no. 4, pp. 287–364, 2012.

[8] Daniel Wolff and Tillman Weyde, "Learning music similarity from relative user ratings," *Information Retrieval*, vol. 17, no. 2, pp. 109–136, 2014.

[9] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[10] Ejaz Ahmed, Michael Jones, and Tim K Marks, "An improved deep learning architecture for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[11] Jiwen Lu, Gang Wang, Weihong Deng, Pierre Moulin, and Jie Zhou, "Multi-manifold deep metric learning for image set classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[12] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun, "Very deep convolutional networks for natural language processing," *arXiv preprint arXiv:1606.01781*, 2016.

[13] Sander Dieleman, Philémon Brakel, and Benjamin Schrauwen, "Audio-based music classification with a pretrained convolutional network," in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, 2011.

[14] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen, "Deep content-based music recommendation," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2013.

[15] Jan Schluter and Sebastian Bock, "Improved musical onset detection with convolutional neural networks," in *Proceedings of the Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[16] Thomas Grill and Jan Schlüter, "Music boundary detection using neural networks on combined features and two-level annotations," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[17] Sebastian Böck, Florian Krebs, and Gerhard Widmer, "Accurate tempo estimation based on recurrent neural networks and resonating comb filters," in *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[19] Edith Law, Kris West, Michael I Mandel, Mert Bay, and J Stephen Downie, "Evaluation of algorithms using games: The case of music tagging.," in *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, 2009.

[20] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[21] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.