TRANSFER LEARNING IN MIR: SHARING LEARNED LATENT REPRESENTATIONS FOR MUSIC AUDIO CLASSIFICATION AND SIMILARITY

Philippe Hamel, Matthew E. P. Davies, Kazuyoshi Yoshii and Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST), Japan
hamelphi@google.com, {matthew.davies, k.yoshii, m.goto}@aist.go.jp

ABSTRACT

This paper discusses the concept of transfer learning and its potential applications to MIR tasks such as music audio classification and similarity.

In a traditional supervised machine learning setting, a system can only use labeled data from a single dataset to solve a given task. The labels associated with the dataset define the nature of the task to solve. A key advantage of transfer learning is in leveraging knowledge from related tasks to improve performance on a given target task. One way to transfer knowledge is to learn a shared latent representation across related tasks. This method has shown to be beneficial in many domains of machine learning, but has yet to be explored in MIR.

Many MIR datasets for audio classification present a semantic overlap in their labels. Furthermore, these datasets often contain relatively few songs. Thus, there is a strong case for exploring methods to share knowledge between these datasets towards a more general and robust understanding of high level musical concepts such as genre and similarity.

Our results show that shared representations can improve classification accuracy. We also show how transfer learning can improve performance for music similarity.

1. INTRODUCTION

As human beings, we are constantly learning to solve new tasks every day. The way we learn to perform new tasks is influenced by what we know about similar tasks [17].

For instance, let's think of a pianist that wants to learn to play guitar. The musician already has some knowledge of music theory, and knows how to use his motor skills to play the piano. When he learns to play guitar, he will not start from scratch but rather use his prior knowledge on music and motor skills and build on top of it. We can see it as if the musician transfers knowledge between tasks by sharing a common abstract internal representation of music.

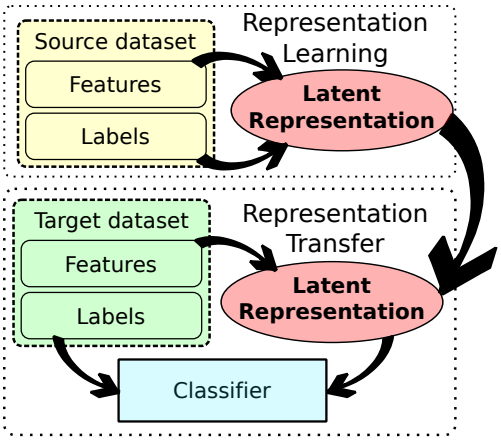


Figure 1: Schema of our transfer learning approach. In the first step, we learn a latent representation in a supervised way using a source dataset. In the second step, we solve the target task by first mapping the features to the learned latent space. In this example, the target task is a classification task.

The equivalent concept in machine learning is called *transfer learning*. It has been applied successfully in many domains such as visual object recognition [13] and webpage classification [8].

The performance of a supervised machine learning system is limited by the quantity and the quality of available labeled data. Obtaining such data can be expensive. As a consequence, many datasets in the MIR community have a relatively small number of labeled examples. Some of these datasets have been built to solve the same task, or similar tasks. For example, there exist many datasets for genre classification, and these datasets exhibit semantic overlap in their labels. However, each individual dataset contains a relatively small number of examples. In this context, it would make sense to try to leverage the information from all these datasets to improve the overall performance. Transfer learning might allow us to do just that.

In this paper, we investigate how transfer learning applied to genre classification, automatic tag annotation and music similarity can be beneficial. We hypothesize that transferring latent representations learned on related tasks can improve the performance of a given task when compared with the original features. Our intuition is that the learned representation will retain some knowledge of the

original task and that this knowledge should make the given task easier to solve.

The paper is divided as follows. We begin with an overview of transfer learning in Section 2. We describe the different MIR tasks that are relevant to our experiments in Section 3. In Section 4 we give details about how we handle our features. The representation learning algorithm is presented in Section 5. We describe our experimental results in Section 6. Finally, we conclude in Section 7.

2. TRANSFER LEARNING

Transfer learning is a machine learning problem that focuses on reusing knowledge learned on one problem in order to help solve another. More formally, we will distinguish between the *target task*, which is the task that we ultimately want to solve, and the *source task* which is the related task that will help us in solving the target task. It is worth noting that there could be more than one source or target task.

Transfer learning is an active field of research, and many approaches have been proposed [2, 8, 13]. Pan et al. [6] describe four transfer learning approaches: i) *instance transfer*, ii) *feature representation transfer*, iii) *parameter transfer* and iv) *relational knowledge transfer*. In this work, we will focus on the feature representation transfer approach, which consists of learning a common feature space between the source and target tasks. More specifically, we will use a supervised approach to construct a feature space using labeled data from a source task, and then use this feature space to help solve the target task. This transfer learning process is illustrated in Figure 1.

Although transfer learning has been applied successfully in many domains, only a few applications can be found in the MIR domain. In [8], self-taught learning, which is an extension of semi-supervised learning, is applied to many tasks, including a 7-way music genre classification. However, very few details are provided on the nature of the music data. In [3], a deep representation learned on genre classification is used for automatic tag annotation. Although the transferred representation is compared to a set of audio features, there is no comparison to the original spectral features that were used to build the deep representation. Thus, it is difficult to assess the impact of the transfer of representation. In [10], a learned automatic tag annotation system is used to produce features to help solve a music similarity task. In [15], a method that attempts to capture the semantic similarities between audio features, tags, and artists names is presented. This multitask approach consists of embedding the different concepts in a common low-dimensional space. This learned space can then be used to solve many MIR related tasks. In our work, we use a similar approach to build a shared latent representation.

3. TASKS AND DATASETS

In this paper, we investigate transfer learning over three related MIR tasks: genre classification, music similarity estimation and automatic tag annotation. Even though these task all use music audio as input data, they differ in their goal and in the way performance is evaluated.

Table 1: Characteristics of the genre classification and automatic tag annotation datasets.

Dataset	# of excerpts	# of classes	Audio length
1517-Artists [11]	3180	19	full
GTZAN [14]	1000	10	30s
Homburg [4]	1886	9	10s
Unique [12]	3115	14	30s
Magnatagatune [5]	22787	160	30s

Table 2: Genre classes for the datasets. In bold are the terms which are also tags in the Magnatagatune dataset [5].

1517-Artists	GTZAN	Homburg	Unique
Alternative & **Punk**	**blues**	alternative	**blues**
Blues	**classical**	**blues**	**country**
Childrens's	**country**	**electronic**	**dance**
Classical	disco	**folkcountry**	**electronica**
Comedy & Spoken Word	**hiphop**	**funk**soulrnb	**hip-hop**
Country	**jazz**	**jazz**	**jazz**
Easy Listening & **Vocals**	**metal**	**pop**	**klassik**
Electronic & **Dance**	**pop**	**raphiphop**	reggae
Folk	reggae	**rock**	**rock**
Hip-Hop	**rock**		schlager
Jazz			soul_rnb
Latin			volksmusik
New Age			**world**
R&B & Soul			wort
Reggae			
Religious			
Rock & **Pop**			
Soundtracks & More			
World			

3.1 Genre Classification

Genre classification consists of choosing the genre that best describes an audio excerpt given a set of genre labels. We consider 4 different datasets for genre classification: 1517-Artists [11], GTZAN [14] Homburg [4], and Unique [12]. These datasets each contain between 1000 and 3180 audio excerpts, from 10 seconds in length to full songs, classified in 9 to 19 genres. In the case where full songs are provided, we use only the first 30 seconds of each song. Further details about the datasets are in Table 1. The genre labels have strong semantic overlap across datasets as can be seen in Table 2. For simplicity, we will sometimes refer to the 1517-Artists dataset as *artists*.

To evaluate the performance of a genre classification system, we use the classification accuracy, which is simply the percentage of correctly classified excerpts in the test set.

3.2 Music Similarity

Music similarity systems seek to obtain a measure of similarity between audio excerpts.

One issue with this task is that the meaning of *similarity* is ill-defined. What is considered similar by one listener might not be the same for another. Another issue is that

similarity is a pair-wise relative measure. Thus, it is complicated and costly to obtain enough ground truth information from human listeners to fully evaluate music similarity systems. In order to circumvent these issues, music similarity systems often use genre labels as a proxy for similarity evaluation [7, 9, 12]. In this context, we consider that two excerpts within the same genre must be more similar than two excerpts from different genres. On this basis, we will use the same datasets as for genre classification in our music similarity experiments.

Even though genre classification and music similarity use the same data, the tasks differ on how we use the data and on how we evaluate performance. Typically, in the music similarity literature [7, 9, 12], the labels are not used for training. Thus, the task must be solved by signal processing, unsupervised learning, or, in our case, by transferring supervised learning from external datasets.

The evaluation of music similarity systems typically use *precision at* k as a performance measure. *Precision at* k gives the ratio of excerpts of the same class in the k nearest neighbors of a given excerpt. In this work we use $k = 10$.

Approaches to solve this task typically consist of measuring distances in a feature space to obtain a distance matrix. The type of features and the distance measure used can vary. In [7], distance is computed using the Jensen-Shannon divergence on a Gaussian representation of the features. In [12], an L_1-distance is computed over aggregated block-level features. In [9], an L_1-distance is computed on features extracted in an unsupervised fashion.

In this work, we use the L_1-distance on our different feature sets in order to obtain a similarity matrix. We also tested Euclidian distance and Cosine distance and obtained similar results.

3.3 Tag annotation

The automatic tag annotation task consists of assigning words to describe an audio excerpt. It is a multi-label problem, meaning that many labels can be applied to a single example. In this paper, we use the Magnatagatune dataset [5] which contains more than 22,000 30-seconds excerpts and 160 tags. Tag labels include musical genre (rock, blues, jazz), instrumentation (guitar, piano, vocals), mood (sad, mellow), other descriptors (fast, airy, beat), etc. There is high semantic overlap with the genre labels from the four genre datasets. We illustrate this, in Table 2, by putting in bold the genres which are also tags in the Magnatagatune dataset.

4. AUDIO FEATURES

In our experiments, we extract Mel-spectrum features from audio. We compute the Discrete Fourier Transform (DFT) on frames of 46ms (1024 samples at 22kHz sampling rate) with half frame overlap. We then pass the magnitude spectrum through 200 triangle Mel-scaled filters and take the log-amplitude to obtain the Mel-spectrum features. These are what we will refer to as *frame-level* features.

However, frame level features have been shown to be suboptimal for genre classification [1]. To obtain a better classification performance, we aggregate features on windows of 64 frames (about 1.5s), computing the mean, variance, maximum and minimum of each feature. We can apply this aggregation process to the Mel-spectrum features as well as to the frame-level latent representations. We will refer to aggregated features as *window-level* features.

5. LEARNING A LATENT REPRESENTATION

In order to transfer knowledge between tasks, we aim to learn a latent representation that will be shared across tasks. To learn this representation, we use the linear embedding method described in [16]. This method consists of embedding both the features and the labels via linear transformations in a common space. This algorithm is built to handle a large number of labels in a multi-label problem, such as in the case of automatic tag annotation. However, the model can trivially be adapted to multi-class problems with a small number of classes such as genre recognition. The model has also been extended to multi-task learning in MIR in [15].

The algorithm seeks to map both the features and the labels in a common latent space, as illustrated in Figure 2. Given a feature representation $x \in \mathbb{R}^d$ and a set of labels $i \in \mathcal{Y} = \{1, ..., Y\}$, we seek to jointly learn a *feature embedding transform* that will map the feature space to a semantic space $\mathbb{R}^D$

$$\Phi_x(x) : \mathbb{R}^d \rightarrow \mathbb{R}^D$$

and a *label embedding transform* that will map labels to the same semantic space

$$\Phi_y(i) : \{1, ..., Y\} \rightarrow \mathbb{R}^D.$$

Thus, in this latent space, it is possible to measure distances between different concepts such as between two feature vectors, a feature vector and a label, or between two labels.

Since we use linear maps, we have $\Phi_x(x) = Vx$ where V is a $D \times d$ matrix and $\Phi_y(i) = W_i$ where W_i is the i-th column of a $D \times Y$ matrix. We can obtain an affinity measure between a feature vector and a given label with

$$f_i(x) = \Phi_y(i)^\top \Phi_x(x) = W_i^\top Vx.$$

Each training example has positive and negative labels associated to it. Given a feature vector, an optimal representation would yield high affinities for positive labels and low affinities for negative labels. In other words, if we rank the affinities of the labels to the feature vector, the positive labels should be ranked low (i.e. in the first few positions), and the negative labels should be ranked high. Computing the exact ranking of the labels becomes expensive when there are many labels. Thus, following [16] we use a stochastic method that allows us to compute an approximate ranking.

The training procedure is as follows. For a given training example x', we randomly pick a positive label j. Then,

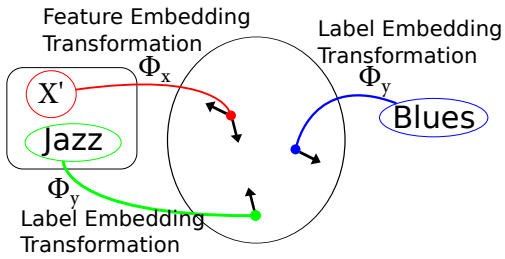


Figure 2: Illustration of the learning of the latent representation. Audio features and labels are mapped to a common embedding space via the transformations Φ_x and Φ_y. In this example, excerpt X' has *jazz* as a positive label and *blues* as a negative example. The black arrows illustrate how the learning gradient will push the negative label embedding and the feature embedding away from each other, while pulling the positive example embedding and the feature embedding together.

we iterate randomly through the negative labels until we find a label j' for which $f_{j'}(x') > f_j(x') - 1$. If we do not find such a negative label, we move to the next training example. If we only need a few iterations to find such a negative label, chances are that the rank of the positive label is high, we thus try to correct this by boosting the loss. On the contrary, if we need many iterations to find such a negative label, the rank of the positive label is probably quite low, so we do not need to change the representation as much. We then minimize the loss given by

$$\mathcal{L} = L(r)|1 - f_j(x') + f_{j'}(x')|$$

where $L(r) = \sum_{k=1}^{r} 1/k$ and r is the approximated rank of the label j and is given by

$$r = \left\lfloor \frac{Y-1}{N} \right\rfloor$$

where N is the number of iterations needed to find j', and $\lfloor \cdot \rfloor$ is the floor function. The loss $\mathcal{L}$ is known as the Weighted Approximate-Rank Pairwise loss, or WARP loss [16]. The L term increases as the approximate rank r grows. The second term in the loss can be seen as a kind of hinge loss, which tries to maximize the margin. For a more in depth description of the algorithm, see [16] and [15] .

In our experiments we used a batch method, meaning that we average the gradient over a batch before updating the parameters. We use 100 examples per batch. For the dimensionality of our latent space, we followed [16] and [15] and chose $D = 100$ as the latent dimensionality for all our experiments.

To extend the model to a multi-dataset setting, we simply alternate between datasets after each batch. The feature embedding transformation is shared across all datasets, but the label embedding transformations are independent. In this way, we do not assume any semantic similarity between similar classes across datasets. In Section 6.1, we show that the model naturally learns these semantic similarities.

6. EXPERIMENTS

We conduct several experiments to assess if transferring knowledge across datasets and task can be beneficial. First, we qualitatively evaluate the semantic similarity in a multi-dataset genre embedding. Then, we compare genre classification performance between tag embedding, genre embedding and the base features. Finally, we use these feature spaces for the music similarity task.

6.1 Semantic similarity

In our first experiment, we learn an embedding jointly on the four genre datasets. The combination of the four label sets gives us a total 52 labels. We then look at the nearest neighbours of the class embedding and make a qualitative evaluation. If the embedding process learns semantic information about the classes as expected, similar classes across datasets should be close to each other.

To do this, we compute a distance matrix using an L_1-distance on the embeddings of all the classes. Then, for each class, we look at which classes are the closest and perform a qualitative evaluation. Some typical examples are presented in Table 3. In general the similar classes across datasets tend to be close to one another. For example, in Table 3, we see that the *jazz* classes all end up near one another in the embedding space. However, there are also some problematic classes. For instance, the *blues* classes do not appear to all be clustered together. From these results, we can say that the embedding space indeed learns some kind of semantic knowledge about the classes.

6.2 Genre Classification

For this experiment, we consider three sets of features for each genre dataset: base features, genre embedding and tag embedding. The base features are the window-level aggregated Mel-spectrum features described in Section 4.

For a given genre dataset, the genre embedding is learned jointly on the 3 other genre datasets. It is learned on frame-level Mel-spectrum features. The frame-level embedded features are then aggregated in a similar fashion as the base features to obtain window-level features.

The tag embedding is learned on the Magnatagatune dataset. Again, the embedding is learned on frame-level features and these are then aggregated to obtain window-level features. We then train a simple linear regression classifier on the window-level features. Finally to classify a song, we average the output of the classifier on the whole song and pick the class with the highest output.

One of the key strengths of transfer learning compared to standard learning is the ability to improve performance using fewer training examples [8]. To test this hypothesis, we measure the accuracy of the classifier across a range of training examples per class in the target dataset. Since the number of examples per class is unbalanced in some datasets, there are cases where there are fewer examples for the less frequent classes. We ran a 10-fold cross-validation for each experiment. The results are shown in Figure 3.

Table 3: Nearest neighbouring classes in the genre embedding space for a few examples.

Seed	5 Nearest neighbours (in order)
Hip-Hop(artists)	hip-hop(unique), raphiphop(homburg), schlager(unique), hiphop(gtzan), Electronic & Dance(artists)
Rock & Pop(artists)	rock(unique) rock(homburg) Alternative & Punk(artists) metal(gtzan) alternative(homburg)
Electronic & Dance(artists)	raphiphop (homburg) reggae(unique) electronica(unique) pop(gtzan) dance(unique)
country(gtzan)	country(unique), folkcountry(homburg), rock(unique), Country(artists), Religious(artists)
jazz(homburg)	jazz(unique), jazz(gtzan), Jazz(artists), world(unique), dance(unique)
blues(unique)	alternative(homburg), Alternative & Punk(artists), blues(gtzan), funksoulrnb(homburg), rock(homburg)

Table 4: Classification accuracy and standard error on the full training set using a 10-fold cross-validation.

Dataset	Base Features	Genre Embedding	Tag Embedding
Artists	0.323 +/- 0.010	0.310 +/- 0.005	0.338 +/- 0.007
GTZAN	0.748 +/- 0.010	0.671 +/- 0.014	0.754 +/- 0.015
Homburg	0.580 +/- 0.012	0.561 +/- 0.009	0.584 +/- 0.008
Unique	0.651 +/- 0.006	0.634 +/- 0.005	0.666 +/- 0.006

Table 5: Precision at 10 for the music similarity task on different feature spaces. The genre embedding is learned using the 3 other genre datasets. The tag embedding is learned on the Magnatagatune dataset.

Dataset	Base Features	Genre Embedding	Tag Embedding
Artists	0.15	0.19	0.19
GTZAN	0.48	0.52	0.53
Homburg	0.36	0.41	0.40
Unique	0.53	0.52	0.54

These results show that the tag embedding often significantly outperforms the base features. This confirms our hypothesis. However, the genre embedding does not perform as well, obtaining better accuracy only for the *Homburg* dataset.

We then measured the accuracy of the three feature sets on the full training dataset. The results are in Table 4. We see that the tag embedding tends to give slightly better results.

6.3 Music similarity

For this task, we used the same 3 feature sets as in Section 6.2. We use precision at 10 as the performance measure. Results are shown in Table 5. We see that both the genre and tag embedding features perform better than the base features, except for the Unique dataset where the three feature sets perform about as well.

7. CONCLUSION

In this paper, we conducted experiments on sharing a learned latent representation between related MIR tasks. We showed that jointly learning a representation on many genre datasets naturally learns semantic similarity between genre classes. In the context of genre classification, we saw that transferring a representation between tasks can significantly improve classification accuracy when the number of training examples is limited. In the context of music similarity, we saw that the similarity space obtained by embedding features using genre and tag labels allows better precision.

The fact that the genre embedding performed worse than the base features for the genre classification task goes against our hypothesis that classification accuracy should be improved by such a representation. This might be due to the fact that the genre datasets are rather small, and thus there was not enough data to learn a robust representation. Another reason might be that some of the semantic knowledge that was learned ended up in the label embedding transform rather than the feature embedding transform. Since we did not use the label embedding transform in the classification task experiment, some of the learned knowledge might have been lost in the transfer. To address this problem in future work, we could try to impose a more severe regularization on the label embedding transformation in the learning process. This could help to force the semantic knowledge to go in the feature embedding transformation.

In this work, to focus on the simplest case first, we limited ourselves to basic feature aggregation, a linear embedding method, and a linear classifier. Each of these elements could be improved further. Thus the performance measures presented in this paper might not reflect the full power of transfer learning. For the features, more complex block-level features as described in [12] could be constructed from the learned frame-level representation. For the representation learning, non-linear mappings could be used to obtain a more powerful representation. Finally, more complex classifiers, such as support vector machines or neural networks could be used to improve classification accuracy on the learned features.

This work presents a first analysis of the potential of transfer learning in MIR. We hope that the results presented here will stimulate more research in the field and motivate the application of transfer learning in future MIR applications.

8. ACKNOWLEDGMENTS

This work was supported by OngaCREST, CREST, JST.

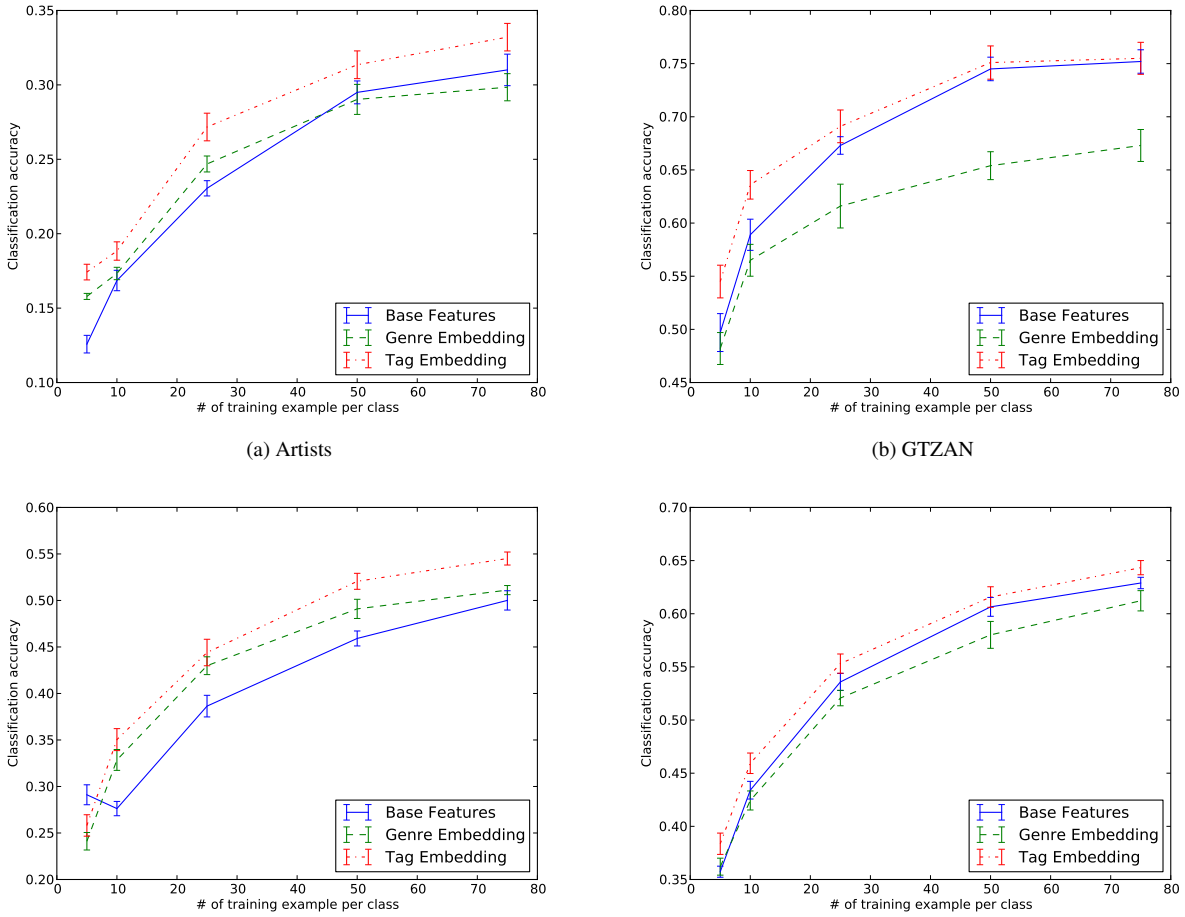
9. REFERENCES

[1] J. Bergstra. Algorithms for Classifying Recorded Music by Genre. Masters thesis, Université de Montréal, 2006.

[2] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, July 1997.

[3] P. Hamel and D. Eck. Learning features from music audio with deep belief networks. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, pages 339–344, 2010.

[4] H. Homburg, I. Mierswa, B. Mller, K. Morik, and M. Wurst. A benchmark dataset for audio classification and clustering. In *Proceedings of the 6th International Society for Music Information Retrieval Conference (ISMIR)*, pages 528–531, 2005.



(a) Artists

(b) GTZAN

(c) Homburg

(d) Unique

Figure 3: Comparison of base features (baseline) to genre embedding and tag embedding for the genre classification task. The genre embedding and tag embedding representations are obtained through our proposed transfer learning method. The error bars correspond to the standard error across the 10 folds.

[5] E. Law and L. von Ahn. Input-agreement: a new mechanism for collecting data using human computation games. In *Proceedings of the International Conference on Human factors in computing systems*, pages 1197–1206, 2009.

[6] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October 2010.

[7] T. Pohle, D. Schnitzer, M. Schedl, P. Knees, and G. Widmer. On rhythm and general music similarity. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pages 525–530, Kobe, Japan, 2009.

[8] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the Twenty-Fourth International Machine Learning Conference (ICML 2007)*, pages 759–766, Corvallis, Oregon, USA, 2007.

[9] J. Schlüter and C. Osendorfer. Music Similarity Estimation with the Mean-Covariance Restricted Boltzmann Machine. In *Proceedings of the 10th International Conference on Machine Learning and Applications (ICMLA 2011)*, pages 118–123, Honolulu, USA, 2011.

[10] K. Seyerlehner, R. Sonnleitner, Schedl, D. M., Hauger, and B. Ionescu. From improved auto-taggers to improved music similarity measures. In *Proceedings of the 10th International Workshop on Adaptive Multimedia Retrieval (AMR 2012)*, Copenhagen, Denmark, 2012.

[11] K. Seyerlehner, G. Widmer, and P. Knees. Frame-level audio similarity - a codebook approach. In *Proceedings of the 11th International Conference on Digital Audio Effects (DAFx-2008)*, pages 349–356, Espoo, Finland, 2008.

[12] K. Seyerlehner, G. Widmer, and T. Pohle. Fusing block-level features for music similarity estimation. In *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-2010)*, pages 528–531, Graz, Austria, 2010.

[13] T. Tommasi, N. Quadrianto, B. Caputo, and C. H. Lampert. Beyond dataset bias: Multi-task unaligned shared knowledge transfer. In *Proc. of the 11th Asian Conference on Computer Vision (ACCV)*, pages 1–15, Daejeon, Korea, 2012.

[14] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[15] J. Weston, S. Bengio, and P. Hamel. Multi-tasking with joint semantic spaces for large-scale music annotation and retrieval. *Journal of New Music Research*, 40(4):337–348, 2011.

[16] J. Weston, S. Bengio, and N. Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI*, volume 3, pages 2764–2770, 2011.

[17] R. S. Woodworth and E. L. Thorndike. The influence of improvement in one mental function upon the efficiency of other functions. *Psychological Review*, 8(3):247–261, May 1901.