RANK-BASED LOSS FOR LEARNING HIERARCHICAL REPRESENTATIONS

Inês Nolasco[1], *Dan Stowell*[2]

[1] Centre for Digital Music (C4DM), Queen Mary University of London, London, UK
[2] Tilburg University, Tilburg, The Netherlands; Naturalis Biodiversity Centre, Leiden, The Netherlands

ABSTRACT

Hierarchical taxonomies are common in many contexts, and they are a very natural structure humans use to organise information. In machine learning, the family of methods that use this ”extra” information is called hierarchical classification. However, applied to audio classification, this remains relatively unexplored. Here we focus on how to integrate the hierarchical information of a problem to learn embeddings representative of the hierarchical relationships. Previously, triplet loss has been proposed to address this problem, however it presents some issues like requiring the careful construction of the triplets, and being limited in the extent of hierarchical information it uses at each iteration. In this work we propose a rank based loss function that uses hierarchical information and translates this into a rank ordering of target distances between the examples. We show that rank based loss is suitable to learn hierarchical representations of the data. By testing on unseen fine level classes we show that this method is also capable of learning hierarchically correct representations of the new classes. Rank based loss has two promising aspects, it is generalisable to hierarchies with any number of levels, and is capable of dealing with data with incomplete hierarchical labels.

Index Terms— hierarchical data, metric learning, rank, embeddings

1. INTRODUCTION

Many supervised learning tasks can be framed as hierarchical problems, meaning that the taxonomy that organises the label space can be constructed as to follow a hierarchical tree structure. Instead of having a flat single level label space, in a hierarchical tree structure the labels are organised in different levels and there is a hierarchical relationship between them. Two important characteristics of these taxonomies are that each child label only has one parent, and it is expected that children from the same parent to share a closer similarity (concept and features) with each other than with labels from other parents. Furthermore this similarity increases as we go down in the hierarchy towards the leaf-labels.

Hierarchical classification is supervised learning that makes use of this kind of label structure to not only make better predictions at the leaf-level task, but also to generate predictions at all the other levels of the taxonomy. This approach is attractive in the sense that we build models that give us a complete picture of how the objects are organised and how they are related with other classes. Furthermore it offers the possibility of classifying items of unknown ”leaf” classes into a broad category.

In this work we propose a novel loss function that uses the partial ordering implied by the hierarchical label taxonomy to derive target distances between embeddings and thus learn hierarchical meaningful embeddings, *i.e.,* embeddings that represent the hierarchical relationships in the taxonomy. Unlike other methods, rank based Loss (RbL) is suitable to learn hierarchical taxonomies with any number of levels and conceptually it is also appropriate to deal with examples with incomplete labels, where we may know higher order classes but may be missing the more specific classes.

Our main contributions are: 1) the rank-based loss and 2) preliminary analysis on how RbL behaves in different settings and datasets and how this compares to a quadruplet loss function that integrates hierarchical information proposed in [1]. The remaining of this paper is organised as: Related work in sec.2, in sec.3 the proposed rank based loss is defined, followed by the description of the datasets, feature extraction and network architecture used, evaluation approach and finally experiments. In Sec.4 we present the results and a discussion. Section 5 we conclude this work with a reflection for future directions and improvements.

2. RELATED WORK

While the concept of hierarchical classification is not new, the application to audio data and within a deep learning framework has not been fully explored. The works in [2] and [3] employ the classical method of hierarchical classification based on training separate models for each level of the hierarchy. The predictions at the fine-level classes are simply the result of combining the prediction probabilities at the higher levels.

978-1-6654-0540-9/22/$31.00 ©2022 Crown

From the literature, current non-classical approaches to hierarchical classification are mainly two: Multi-task learning, in which we have one task per level of the hierarchy and following the multi-task learning framework, the network learns to optimise for the multiple tasks together. In this vein, the work in [4] deals with acoustic scene classification on a 2 level tree label structure. Besides designing an objective function that combines together the loss at the fine level and coarse level, the authors also propose a scheme of pretraining the networks on a single level of the label tree in order to improve the training and performance on other levels. Another pertinent example is the work in [5] that focuses on bird species classification from flight calls. Here the data is organised in a 3 level taxonomy (animal order, family, species). The authors propose a novel network architecture, the *Taxonet* that is both a hierarchical and multi-task neural network. By partitioning the layers and defining conditional activation of each node given which partition was activated in the previous layer, they are able to translate the hierarchical taxonomy of the problem into the network architecture. Both these examples report improvements using their hierarchical multi-task methods when compared against flat classification, *i.e,.* classification only at the fine level. The other main approach for hierarchical classification is more related with metric learning and generating embeddings that explicitly convey the hierarchical structure of the problem. The core idea is that distances between similarly labelled examples should be minimised and examples more distant in the label space should have their distances maximised. With this intent, in [6], the authors explore the use of Siamese networks and manually define a target distance between pairs of items in the generated embeddings depending on the position of the input examples on the label tree. Hierarchical information is also integrated through the network architecture by multiplying an incidence matrix with the output layer predicting the leaf-level of classes which generated the output predictions for the higher level classes. In [7], the authors address musical instrument recognition in a few shot setting. They use prototypical networks to generate embeddings at both levels of a music instrument hierarchy, by aggregating the leaf-level embeddings according to the label structure of the problem. Another relevant work [1] employs a quadruplet loss (generalisation of triplet loss) approach to sound event detection on a dataset organised in a two level hierarchical tree. The core of their proposed method is to build quadruplets that contain examples of all the possible hierarchical relationships of the label tree. *I.e.,* anchor and positive are examples from the same leaf-label, negative are examples from the same coarse level and different leaf-label, or from a different coarse level. The authors report improved classification results at both levels.

Loosely related with learning hierarchical relationships is also the concept of learning to rank. In learning to rank the goal is to learn a function that given a query example will score relevant/closer results higher than irrelevant ones or simply sort a list of results by the relevance to the query. In [8] the authors frame the problem of learning to rank as a metric learning problem, and propose a method that directly optimises for ranking. Also, in [9] the concept of rank is explored connected to a classical hierarchical classification approach: the authors propose the training of leaf-level classifiers weighting more data examples that are closest in ranking to each class being trained.

3. METHODS

3.1. Rank-based loss

Our proposed loss function[1] follows a metric learning approach, where the objective is to learn embeddings in which the distances between them are meaningful to the problem being addressed. Similarly to the quadruplet loss proposed in [1], at each iteration of training we want to evaluate the distances between embeddings and push the embeddings closer or further away depending on the hierarchical relationship between labels. The hierarchical information is used here to define, for each element, the desired rank-ordering of all other elements in terms of their distance. For each pair of embeddings we compute a loss value that is either 0 if the pair has a "correct" distance given the rank, or a positive value meaning how far away from the target distance the pair is. Formally the rank based loss is defined as:

$$L = \frac{1}{P}\sum_{p}^{P}(1-I_p).(EmbDist_p - TargetDist_p)^2 \quad (1)$$

where $EmbDist_p$ is the cosine distance in the embedding space between two embeddings. $TargetDist_p$ is the target distance for that pair given the desired rank of the pair. I_p is a Boolean indicating if the pair is correctly distanced given their rank in the label tree.

We summarise computation of this loss in 5 steps:

1) Compute a rank map from the tree of ground truth labels: Each pair of examples has a rank given by the tree distance of their labels. The tree distance is given by the number of nodes that separate the two labels

2) Compute all the pairwise cosine distances in the batch in the embedding space, and sort them.

[1] The Pytorch implementation of Rank based loss is available: https://github.com/inesnolas/Rank-based-loss_ICASSP22

3) For each rank, assign a target distance by selecting whatever distance in the sorted distances vector falls at each rank.

4) Compute I_p as: 0 if distance of the pair is within the correct positions in the sorted distances vector, else 1 if distance of the pair is wrong given the ground truth rank.

5) Compute the loss from eq.1.

3.2. Datasets

Three datasets from different contexts were used:

3 Bird species, audio data collected to accompany the work in [10] on automatic acoustic identification of individual animals. It contains labelled recordings of individuals from three different bird species: Little owl (Athene noctua), Chiffchaff (Phylloscopus collybita), and Tree pipit (Anthus trivialis). For this work, the data was augmented by mixing foreground recordings of each individual with background recordings of other individuals. Furthermore we re-structured the labels to follow a 3 level hierarchical taxonomy: taxonomic group, species, and individual identity. From this dataset, a total of 1707 recordings were selected belonging to 9 individuals equally distributed across the 3 species of birds.

Nsynth, a large-scale dataset of annotated musical notes [11], it contains 4-second audio snippets of notes played with different instruments. For this work a small selection of the dataset was created to address the task of instrument recognition. A 2-level hierarchical taxonomy is built using the instrument family labels as highest level classes, and the instrument id as the fine-level. We selected a total of 1707 audio snippets from 9 instruments across the guitar, flutes and keyboard families. All instruments are from the "acoustic" source.

TUTasc2016, dataset of 30-second audio segments from 15 acoustic scenes [12]. These are organised in 3 groups: indoor, outdoor and vehicles accordingly to the environment where they were captured. For the hierarchical taxonomy we consider the acoustic scene labels as the fine level classes and the 3 groups as the coarse level classes. The selected data used in this work consists of 704 recordings coming from 9 acoustic scenes, balanced across the 3 groups: library, home, and metro station from indoors; tram, bus and train from vehicles; residential area, forest path and beach from outdoors.

3.3. Feature extraction and architecture

From the raw audio recordings sampled at 16kHz sample rate, we compute log mel spectrograms with 64 frequency bands, a window length of 400 samples and hop size of 160 samples. These spectrograms are then passed through a VGGish network [13] previously trained on the Audioset dataset, that generates one 128-dimensional embedding vector for each second of the log mel spectrograms. We use an openly-available pytorch implementation[2]. Additionally the generated embeddings are averaged over time in order to obtain a single embedding vector for each recording. Before being fed into the network these embeddings are standardised based on the mean and standard deviation of the training set.

The trainable network where the loss function is to be tested consists on a single linear layer that receives the 128 dimensional embedding and transforms it into a 3 dimensional embedding vector of norm 1.

3.4. Evaluation

For evaluation purposes we want to focus on the quality of the embeddings learned, and how well these can express the hierarchical structure of the problems. For that purpose we compute the silhouette score[14] based on the ground truth labels. This score is computed by averaging the silhouette coefficient across all the samples in the set:

$$Sil = \frac{1}{N}\sum_{n}^{N}\frac{b-a}{\max(a,b)}, \tag{2}$$

Where a is the intra-cluster distance and b is the mean nearest-cluster distance. This metric expresses how well the samples are positioned accordingly to the ground truth clusters, it has values ranging between 1 and -1, from the best case where all samples are positioned within the correct cluster to the worst, where samples are positioned as if belonging to the wrong cluster. A value close to zero means that the clusters are overlapping and are difficult to separate. For each set we compute the score using the labels at the fine level, and at the coarse level of the hierarchy, thus obtaining a measure of the quality of the learnt embeddings across the hierarchy.

For the 3 datasets described before, we partition the pre-selected data into training set (70%), validation set (20%) and test set(10%) at random. Furthermore, for the majority of the experiments we use predefined batches that contain examples of pairs from all the ranks. This choice is based on the idea that balanced batches in terms of ranks would work better for the rank based loss, and it also makes comparisons with the quadruplet loss more complete. For all the experiments we employ an early stopping procedure (patience of 20) based on the average of silhouette scores in the validation set. *i.e* the training stops once the max value for the averaged silhouette score is reached. Furthermore, with the purpose of testing the capability of the models to generalise to unseen classes at the fine level, additional test sets were created from different fine classes than the development sets,

[2] https://github.com/harritaylor/torchvggish

		3 bird species			Nsynth			TUTasc2016		
		Sil^{Fine}	Sil^{Coarse}	$avSil$	Sil^{Fine}	Sil^{Coarse}	$avSil$	Sil^{Fine}	Sil^{Coarse}	$avSil$
Test	**InitEmb**	**-0.07 (0.0)**	0.19 (0.0)	0.06 (0.0)	**0.02 (0.0)**	0.29 (0.0)	0.16 (0.0)	-0.02 (0.0)	0.18 (0.0)	0.08 (0.0)
	QuadL	-0.17 (-0.10)	0.32 (+0.13)	0.07 (+0.01)	0.01 (-0.01)	**0.60 (+0.31)**	**0.31 (+0.15)**	-0.19 (-0.17)	0.14 (-0.04)	-0.02 (-0.10)
	RbL	-0.09 (-0.02)	**0.42 (+0.23)**	**0.17 (+0.11)**	-0.08 (-0.10)	0.46 (+0.17)	0.19 (+0.03)	0.03 (+0.05)	0.60 (+0.42)	0.31 (+0.23)
	RbL_unc	-0.23 (-0.16)	0.23 (+0.04)	0.0 (-0.06)	-0.16 (-0.18)	0.38 (+0.09)	0.11 (-0.05)	**0.06 (+0.08)**	**0.73 (+0.55)**	**0.40 (+0.32)**
NovelTest	**InitEmb**	**-0.06 (0.0)**	0.35 (0.0)	0.14 (0.0)	**-0.02 (0.0)**	0.04 (0.0)	0.01 (0.0)	0.17 (0.0)	0.22 (0.0)	0.19 (0.0)
	QuadL	-0.08 (-0.02)	**0.57 (+0.22)**	**0.25 (+0.11)**	-0.09 (-0.07)	-0.02 (-0.06)	-0.05 (-0.06)	0.14 (-0.03)	0.27 (+0.05)	0.21 (+0.02)
	RbL	**-0.06 (0.0)**	0.48 (+0.13)	0.21 (+0.07)	-0.04 (-0.02)	**0.13 (+0.09)**	**0.04 (+0.03)**	0.16 (-0.01)	**0.8 (+0.58)**	0.48 (+0.29)
	RbL_unc	-0.19 (-0.13)	0.32 (-0.03)	0.07 (-0.07)	-0.33 (-0.31)	0.06 (+0.02)	-0.13 (-0.14)	**0.33 (+0.16)**	0.74 (+0.52)	**0.53 (+0.34)**

Table 1: Silhouette scores on the test sets and on the alternative test sets, where the leaf-classes are different from the classes used to train the models.

3.5. Experiments

[InitEmb] Evaluation of initial pretrained embeddings This serves the purpose of defining a baseline for comparison with all other experiments. The main goal is to understand if an improvement over the "non-hierarchical" initial embeddings is achieved or not. The embeddings dimensions were reduced to 3 in order to provide a fairer baseline for comparison.

[QuadL] Quadruplet Loss The loss of [1] is especially relevant to compare with the rank based loss. As mentioned in sec.1, this loss integrates hierarchical information through the selection of the examples that generate the quadruplets. Batch size is 3 (quadruplets).

[RbL] Rank based loss Training the network with the RbL on the 3 datasets. Batch size is 12 and the examples are selected to create a balanced batch across ranks.

[RbL_unc] RbL with unconstrained batches On all the previous experiments the batches are balanced regarding the hierarchical relationships between ground truth labels. Here that constraint is lifted, allowing, for example, batches to be missing pairs of one rank or have a disproportional number of pairs in another.

4. RESULTS AND DISCUSSION

Results are reported in Table 1, giving the fine and coarse level silhouette scores obtained for the test sets. we also report the average between the silhouettes at both levels and to highlight how these compare with the baseline initial embeddings, the difference from the baseline is shown in brackets.

Generally, results show that both rank-based loss and quadruplet loss can learn to represent hierarchical embeddings, but with notable variation across datasets. It is worth noting the difficulty in learning good embeddings to represent the fine level classes. That aside RbL performs well including on novel leaf classes.

Comparing the results for both test sets on the Nsynth and 3 bird datasets, they seem to show a general trend where when the initial representation is worse, RbL scores above the Quadruplet loss. We hypothesise that this is related to the fact that Quadruplet loss defines a lower bound target distance between embeddings that is always different from zero and thus has less "liberty" when moving data around in the embedding space. Another aspect of this is: by defining margins that are not based on the data, quadruplet loss will learn to position embeddings of different classes that are always distanced by the same amount. e.g, the distance between a flute, a guitar and a keyboard are always the same. RbL however, since it gets the target distances from the data, could learn that a guitar and a keyboard are more acoustically close than a flute, and the learnt embeddings express this.

Observing the results for experiment **[RbL_unc]**, as expected the capability of learning hierarchical embeddings drops when we allow the batches to have any composition of examples regarding their hierarchical relationships. We argue however that this is an indication that RbL still allows some flexibility regarding the batch composition and that this is an advantage over the quadruplet loss that requires very strict quadruplet composition. The results for TUTasc2016 are an outlier in which the unconstrained batch performs better, an outcome which merits further study.

5. CONCLUSION

This work presented a novel rank based loss function and we have shown its ability to learn embeddings that are representative of the hierarchy of the labels. Our rank based loss was compared against another loss function that incorporates hierarchical information, with positive results. Next steps involve a more in depth exploration regarding the effect of the dataset structure on the performance of RbL, the use of different distance metrics and evaluation of the approach from classification results. Also it would be interesting to test the RbL with a larger number of hierarchical levels, and show its ability to deal with incomplete labelled data.

6. REFERENCES

[1] Arindam Jati, Naveen Kumar, Ruxin Chen, and Panayiotis Georgiou, "Hierarchy-aware loss function on a tree structured label space for audio event detection," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6–10.

[2] Lam Pham, Ian McLoughlin, Huy Phan, Ramaswamy Palaniappan, and Alfred Mertins, "Deep feature embedding and hierarchical classification for audio scene classification," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–7.

[3] Todor Ganchev and Ilyas Potamitis, "Automatic acoustic identification of singing insects," *Bioacoustics*, vol. 16, no. 3, pp. 281–328, 2007.

[4] Yong Xu, Qiang Huang, Wenwu Wang, and Mark D Plumbley, "Hierarchical learning for dnn-based acoustic scene classification," *arXiv preprint arXiv:1607.03682*, 2016.

[5] Jason Cramer, Vincent Lostanlen, Andrew Farnsworth, Justin Salamon, and Juan Pablo Bello, "Chirping up the right tree: Incorporating biological taxonomies into deep bioacoustic classifiers," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 901–905.

[6] Benjamin Elizalde, Abelino Jimenez, and Bhiksha Raj, "Sound event classification using ontology-based neural networks," in *NIPS 2018 Workshop*, 2018.

[7] Hugo Flores Garcia, Aldo Aguilar, Ethan Manilow, and Bryan Pardo, "Leveraging hierarchical structures for few-shot musical instrument recognition," *arXiv preprint arXiv:2107.07029*, 2021.

[8] Brian McFee and Gert RG Lanckriet, "Metric learning to rank," in *ICML*, 2010.

[9] Azad Naik and Huzefa Rangwala, "A ranking-based approach for hierarchical classification," in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2015, pp. 1–10.

[10] Dan Stowell, Tereza Petrusková, Martin Šálek, and Pavel Linhart, "Automatic acoustic identification of individuals in multiple species: improving identification across recording conditions," *Journal of The Royal Society Interface*, vol. 16, no. 153, pp. 20180940, Apr. 2019.

[11] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi, "Neural audio synthesis of musical notes with wavenet autoencoders," 2017.

[12] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "TUT acoustic scenes 2016, development dataset," Feb. 2016.

[13] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al., "CNN architectures for large-scale audio classification," in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2017, pp. 131–135.

[14] Peter J Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.