Supplementary Material
Hierarchical Learning of Dependent Concepts for Human Activity Recognition

Aomar Osmani[1], Massinissa Hamidi[1](✉), and Pegah Alizadeh[2]

[1] LIPN-UMR CNRS 7030, Univ. Sorbonne Paris Nord, Villetaneuse, France
[2] Léonard de Vinci Pôle Universitaire, Research Center, 92 916 Paris, La Défense
{ao,hamidi}@lipn.univ-paris13.fr, pegah.alizadeh@devinci.fr

Appendix A

Proof. Theorem 1. It can be explained by observing that, for $K+1$ concepts containing K existed concepts $c_1, \cdots c_K$ and a new added concept γ, we can produce the first level trees combinations as below. Notice that each atomic element o can be one of the $c_1, \cdots c_K$ concepts. In order to compute the total number of trees combinations, we show what is the number of tree combinations by assigning the K concepts to each item:

- $(\gamma(\overbrace{o \cdots o}^{K\text{concepts}}))$: the number of trees combinations by taking the concept labels into the account are: $\binom{K}{0}L(1) \times 2 \times L(K)$; the reason for multiplying the number of trees combinations for K concepts to 2 is because while the left side contains an atomic γ concept, there are two choices for the right side of the tree in the first level: either we compute the total number of trees for K concepts from the first level or we keep the first level as a $\overbrace{o \cdots o}^{K\text{concepts}}$ atomics and keep all K concepts together, then continue the number of K trees combinations from the second level of the tree.
- $((\gamma o)(\overbrace{o \cdots o}^{K-1\text{concepts}}))$: similar to the previous part we have $\binom{K}{1}L(2) \times 2 \times L(K-1)$ trees combinations by taking the concepts labels into the account. $\binom{K}{1}$ indicates the number of combinations for choosing a concept from the K concept and put it with the new concept separately. While $L(2)$ is the number of trees combinations for the left side of tree separated with the new concept γ.
- $((\gamma oo)(\overbrace{o \cdots o}^{K-2\text{concepts}})), \cdots$
- $((\gamma \overbrace{o \cdots o}^{K-1\text{concepts}})o)$: $\binom{K}{K-1}L(K)L(1)$ in this special part, we follow the same formula except the single concept in the right side has only one possible combination in the first level equal to $L(1)$.

All in all, the sum of these items calculates the total number of tree hierarchies for $K+1$ concepts.

The first few number of total number of trees combinations for $1, 2, 3, 4, 5, 6, 7, 8, 9, 10, \cdots$ concepts are: 1 , 1, 4, 26, 236, 2752, 39208, 660032, 12818912, $282137824, \cdots$. In the case of the SHL dataset that we use in the empirical evaluation, we have 8 different concepts and thus, the number of different types of hierarchies for this case is $L(8) = 660, 032$.

Appendix B Training Details

We use Tensorflow for building the encoders/decoders. We construct encoders by stacking Conv1d/ReLU/MaxPool blocks. These blocks are followed by a Fully Connected/ReLU layers. Encoders performance estimation is based on the validation loss and is framed as a sequence classification problem. As a preprocessing step, annotated input streams from the huge SHL dataset are segmented into sequences of 6000 samples which correspond to a duration of 1 min. given a sampling rate of 100 Hz. For weight optimization, we use stochastic gradient descent with Nesterov momentum of 0.9 and a learning-rate of 0.1 for a minimum of 12 epochs (we stop training if there is no improvement). Weight decay is set to 0.0001. Furthermore, to make the neural networks more stable, we use batch normalization on top of each convolutional layer. We use SVMs as our ERMs in the derived hierarchies.

Appendix C Evaluation Metrics

In hierarchical classification settings, the hierarchical structure is important and should be taken into account during model evaluation [3]. Various measures that account for the hierarchical structure of the learning process have been studied in the literature. They can be categorized into: distance-based; depth-dependent; semantics-based; and hierarchy-based measures. Each one is displaying advantages and disadvantages depending on the characteristics of the considered structure [2]. In our experiments, we use the *H-loss*, a hierarchy-based measure defined in [1]. This measure captures the intuition that *"whenever a classification mistake is made on a node of the taxonomy, then no loss should be charged for any additional mistake occurring in the sub-tree of that node."* $\ell_H(\hat{y}, y) = \sum_{i=1}^{N} \{\hat{y}_i \neq y_i \wedge \hat{y}_j = y_j, j \in Anc(i)\}$, where $\hat{y} = (\hat{y}_1, \cdots \hat{y}_N)$ is the predicted labels, $y = (y_1, \cdots y_N)$ is the true labels, and $Anc(i)$ is the set of ancestors for the node i.

References

1. Cesa-Bianchi, N., Gentile, C., Zaniboni, L.: Incremental algorithms for hierarchical classification. JMLR **7**(Jan), 31–54 (2006)
2. Costa, E., Lorena, A., Carvalho, A., Freitas, A.: A review of performance evaluation measures for hierarchical classifiers. In: Evaluation Methods for machine Learning II: papers from the AAAI-2007 Workshop. pp. 1–6 (2007)
3. Silla, C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery **22**(1-2), 31–72 (2011)