Clustering Approach to Solve Hierarchical Classification Problem Complexity

Aomar Osmani[1]**, Massinissa Hamidi**[1]**, Pegah Alizadeh**[2]

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

Abstract

In a large domain of classification problems for real applications, like human activity recognition, separable spaces between groups of concepts are easier to learn than each concept alone. This is because the search space biases required to separate groups of classes (or concepts) are more relevant than the ones needed to separate classes individually. For example, it is easier to learn the activities related to the body movements group (running, walking) versus "on-wheels" activities group (bicycling, driving a car), before learning more specific classes inside each of these groups. Despite the obvious interest of this approach, our theoretical analysis shows a high complexity for finding an exact solution. We propose in this paper an original approach based on the association of clustering and classification approaches to overcome this limitation. We propose a better approach to learn the concepts by grouping classes recursively rather than learning them class by class. We introduce an effective greedy algorithm and two theoretical measures, namely cohesion and dispersion, to evaluate the connection between the clusters and the classes. Extensive experiments on the SHL dataset show that our approach improves classification performances while reducing the number of instances used to learn each concept.

Introduction

In Internet of Things and particularly human activity recognition (HAR), concepts (or activities) naturally intermingle and the boundaries are not evident, e.g., the transition from the concept *walking* to *running* remains blurred. These phenomena are accentuated by the sensors capabilities and the perspectives (views) through which the data is collected (position in space, position on the body, sensor characteristics) (Aghajan and Cavallaro 2009; Hamidi and Osmani 2020). Incomplete or redundant perspectives can lead to further confuse the concepts between them and to reduce the performance of the learning process. Beyond the dependencies (overlap) relating to the perspectives provided by the deployments of sensors, the phenomena themselves and the concepts which compose them often exhibit intrinsic dependencies (Silla and Freitas 2011; Essaidi, Osmani, and Rouveirol 2015; Hamidi and Osmani 2021).

We find that some concepts are easier to distinguish when grouped with other concepts than when each one is learned on its own. For instance, if we consider analyzing human activities through the accelerometer and heart rate, it is first easier to separate all activities (concepts) into two main classes, e.g., body required movements versus others movements, instead of separating the activities between these two general classes. This general observation shows that inductive biases needed to separate homogeneous groups of concepts recursively give better results and build hierarchical concept structure between concepts. We propose an original approach for structuring the considered concepts into hierarchies so that very similar concepts are grouped together and tackled by specialized classifiers. The idea is that classification at different levels of the hierarchy may rely on different features or different combinations of the same features (Osmani, Hamidi, and Alizadeh 2021; Yao et al. 2019). Indeed, many real-world classification problems are naturally cast as hierarchical classification problems (Cai and Hofmann 2004; Wehrmann, Cerri, and Barros 2018; Yao et al. 2019; Zhou, Xiao, and Wu 2011; Hamidi, Osmani, and Alizadeh 2020). A work on the semantic relationships between categories in a hierarchical structure shows that they are usually of the type *generalization-specialization* (Zhou, Xiao, and Wu 2011). In other words, the lower-level categories are supposed to have the same general properties as the higher-level categories plus additional more specific properties.

The problem at hand is twice difficult as we have to, first, find the most appropriate hierarchical structure and, second, find optimal learners assigned to the nodes of the hierarchical structure. Some works have tackled this problem by exploiting a priori knowledge and structures of the domain (Samie, Bauer, and Henkel 2020; Scheurer et al. 2020). However, such a priori knowledge is not always available. A naive approach consists in building all the combinations of concepts to check for which groups of classes the quality of the learning is optimal and to start again recursively this approach until the concepts are totally separated from each other. However, this approach faces the combinatorial explosion of the number of cases that should be treated (see § for more details on the complexity analysis).

To overcome this complexity limitation, we propose an original approach combining clustering and classification on groups of concepts based on two original measures between concepts, namely cohesion and dispersion, optimized throughout the process until the derivation of an optimal learning hierarchy. We design a set of training strategies inspired by meta-learning, used to adjust the weights of the learners assigned to the nodes of the hierarchy. The proposed training strategies are specifically designed to leverage the hierarchical structure of the learning process and to reduce the amount of supervision required in low-data regimes. It allows a substantial decrease in the number of required learning examples in order to achieve comparable, sometimes better recognition performances compared to the full-data regime and flat classification setting.

The main contributions of this paper are summarized as follows. (1) We propose two novel measures (dispersion and cohesion) to assess the quality of clustering solutions regarding concepts separability. (2) We propose an efficient clustering-based classification approach combined with training strategies that leverage the tree structure to improve the learning process. The components of the proposed approach, including the theoretical complexity of the hierarchical learning problem, which is substantially reduced, are analyzed. (3) Extensive experiments are conducted on three HAR datasets to assess the effectiveness and efficiency of our proposed approach. The notion of inductive biases inheritance in the hierarchy of concepts being derived is also investigated. Furthermore, we empirically analyze the notion of intrinsic concept dependency and its relativity w.r.t. to the various perspectives (views) provided by the sensors deployments and how the proposed measures capture these two kinds of dependency.

Problem Statement

The main idea of this paper comes from the fact that the concepts to be learned are not totally independent, as is the case in human activity recognition where, for example, learning the concept *running* is closer to learning the concept *walking* than learning the concept *still*. Thus, grouping some concepts to learn them against other groups of concepts, using more adapted biases or characteristics, can considerably improve the learning process quality for each concept.

Let $\mathcal{H}$ be a hypothesis space, $\mathcal{A} = \{A_1, \ldots, A_n\}$ a set of atomic concepts to learn, and an input space $\mathcal{X}$. The mapping between the input space and the set of concepts is described by a set of instances (labelled examples) $\mathcal{E} = \{e_1, \ldots, e_m\}$, where each instance e_i is defined as a couple (X_i, A_i) with $X_i \in \mathcal{X}$ and $A_i \in \mathcal{A}$. The main goal is to find an optimal hypothesis (or theory) $h^* \in \mathcal{H}$ able to explain the instances (or the mapping between the input space and the set of concepts). This goal corresponds to minimizing the empirical risk $R_{emp}(h)$, i.e., $h^* = \text{argmin}_{h \in \mathcal{H}} R_{emp}(h)$, which is computed by averaging a given loss function ℓ on the set of instances as follows: $R_{emp}(h) = \frac{1}{|\mathcal{X}|} \sum_i \ell(h(X_i), A_i)$ [1].

In this paper, we show that for a given specific *a priori* knowledge on the concepts to learn, the quality of the learned hypothesis improves by grouping the concepts recursively. We assume that atomic concepts are not decomposable, i.e., $\forall i \neq j \in \{1, \ldots, n\}, A_i \not\subset A_j$), and any group of concepts GA_i is a subset of $\mathcal{A}$. Since the atomic concepts have partial dependencies in many cases, a top-down approach tries to structure the atomic concepts into different combinations and based on different biases. It gives a better loss function than the one used in the flat case. This idea is close to the decision tree (Quinlan 1986) but more general. It is applied to the separability of the groups of concepts rather than to atomic concepts. This formalization extends the idea presented in (Kosmopoulos et al. 2015) which defines a three dimensions setting: (1) single-label classification as opposed to multi-label classification; (2) concepts are organized into trees as opposed to directed acyclic graphs; (3) instances are classified into leaves (mandatory leaf node prediction (Silla and Freitas 2011)), as opposed to the setting where instances can be classified into any node of the hierarchy. One of the main problems to solve, in this case, is finding the best tree structure of groups of concepts to learn together in order to optimize the learning rate of each atomic concept. However, the complexity of this problem is prohibitive:

Theorem 1. *Let $L(n)$ be the total number of trees for the n atomic concepts. The search space size for these concepts satisfies a recurrence relation defined as:*

$$L(n) = \binom{n-1}{n-2} L(n-1)L(1) + 2\sum_{i=0}^{n-3} \binom{n}{i} L(i+1)L(n-i-1)$$

Proof. See the Supplementary Material. □

Because of the exponential size of the search space, the exact approaches cannot tackle this problem in terms of time/space complexity for large sets of (fine or coarse-grained) concepts like those featured by the SHL dataset (Gjoreski et al. 2018), which we consider throughout the paper as a running example to illustrate the problem and the proposed approach on a concrete real-world example. In this dataset for example, with 8 coarse-grained concepts, the size of the search space is $L(8) = 660,032$.

To take advantage of the power of this search space traversal approach and to avoid combinatorial explosion, we will detail our clustering-based approach for selecting the best concept group structure.

Proposed Approach

In several applications, it is more convenient to consider that the biases used to separate groups of classes is different from those used to separate classes in each group. However, as shown in the previous section, this formulation of the learning problem is prohibitive in its original form. To overcome this limitation, we propose an original approach based on a sequence of recursive clustering steps to guide the choice of appropriate groups of concepts and corresponding learning biases. In this section, we detail the different parts of our approach which are illustrated in Figure 1.

[1] In the HAR considered applications, activity recognition is addressed according the following predefined chain (Bulling, Blanke, and Schiele 2014): the labelled examples generated from the sensors are (1) segmented into short sequences; which are (2) preprocessed; and (3) from which discriminative features are extracted; (4) before being fed into a machine learning algorithm responsible of finding the mapping towards the activities (concepts).

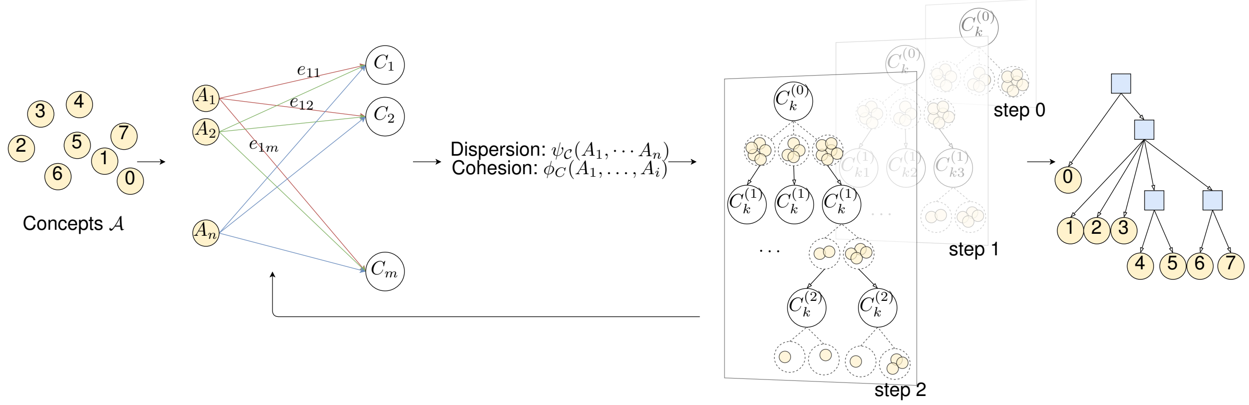

Figure 1: Framework of the proposed approach. Based on the dispersion and cohesion score obtained for each cluster, the best clustering solution is selected (step 0) and the process is repeated recursively on each group of concepts within the selected clustering solution (subsequent steps 1, 2, etc.). The process ends as soon as we get individual concepts on the leaves of the decomposition hierarchy. The final hierarchy is assigned with specialized learners at every non-leaf node, which are trained on the groups of concepts within its descendant leaves.

Clust#	*still*	*walk*	*run*	*bike*	*car*	*bus*	*train*	*subway*	Cohes.
C_1	15	0	3	12	1	3	0	6	0.20
C_2	8	50	45	3	2	2	6	7	0.42
C_3	12	3	1	19	1	0	5	9	0.228
C_4	5	2	2	9	18	20	51	45	0.376
Disp.	0.75	1	1	0.75	1	1	1	1	

Table 1: Distribution of the instances within a clustering solution containing 4 clusters. The corresponding cohesion (cohes.) and dispersion (disp.) scores are depicted.

Dispersion and Cohesion Measures

Let $\mathcal{A}$ be the set of concepts we want to learn on the input space $\mathcal{X}$. Let's also consider $\mathcal{C} = \{C_1, \ldots, C_m\}$, a clustering result (or solution) obtained in an unsupervised setting (using only the input features X_i of the instances). Instances of the same concept may be grouped in distinct clusters of the clustering solution. This clustering result can be represented as $G = (\mathcal{A}, \mathcal{C}, E)$ a bipartite graph whose partition has the parts $\mathcal{A}$ (the classification domain) and $\mathcal{C}$ (the clustering domain), with E denoting the edges of the graph (see Figure 1). Each edge $e_{ij} \in E$ represents the percentage of the instances from the input space $\mathcal{X}$ in class A_i, properly covered by the cluster C_j. As a consequence the basic normalization property holds: $\forall\ 1 \leq i \leq n,\ \sum_{j=1}^{m} e_{ij} = 1$.

Clustering on the Running Example. Let's consider a small subset from the SHL dataset containing 365 instances distributed as follows: *still (a_1)*: 40, *walk (a_2)*: 55, *run (a_3)*: 51, *bike (a_4)*: 43, *car (a_5)*: 22, *bus (a_6)*: 25, *train (a_7)*: 62, *subway (a_8)*: 67. Table 1 illustrates the distribution of the instances within a single clustering solution.

We define two measures, namely *dispersion* and *cohesion* between clusters and classes. The main idea is to evaluate how a given clustering result (obtained in an unsupervised manner) captures the dependencies between the considered concepts (or the assigned labels according to the labeling process used in the dataset). The goal is to use the clustering result to select the best groups of concepts in a way that they can be separated effectively by means of concepts cohesion and dispersion. The subsequent learning steps are applied on these groups of concepts.

Definition 1 (Dispersion ψ)**.** *The dispersion of a class A_i related to a cluster C_j denoted as ψ_{ij} defines how the cluster C_j represents the class A_i. ψ_{ij} measures the distribution of the instances labeled as A_i in the $C_j \in \mathcal{C}$ clustering.*

There exist different approaches for defining the class distribution in each cluster. However, in this paper, we consider the basic one $\psi_{ij} = e_{ij}$. We denote this distribution with ψ_{ij}. Consequently, the A_i instances distribution w.r.t. the $\mathcal{C}$ clustering should satisfy the following boundaries for the worst and the best cases:

$$\psi_{\mathcal{C}}(A_i) = \begin{cases} 0 & \text{if } \forall j \in \{1, \ldots, m\}\ \psi_{ij} < \frac{1}{m} + \epsilon \\ 1 & \text{if } \exists j \in \{1, \ldots, m\}\ \psi_{ij} \geq 1 - \epsilon \end{cases}$$

where ϵ is the given small value. If for each class A_i, $\psi_{\mathcal{C}}(A_i) = 0$, then the *dispersion* is total and no cluster represents this class. However, if $\psi_{\mathcal{C}}(A_i) = 1$ because of $\psi_{iw} \sim 1$ then the cluster C_w represents totally the class A_i.

The *dispersion* measure can be computed in several ways. In fact, if a given concept is represented by a clustering solution, it can be measured using statistical or *a priori* known properties. We propose here a simple measure defined as follows. Assume a class A_i and a cluster C_j in the clustering result $\mathcal{C}$, for a given threshold α are given. We define R as an auxiliary measure for the *dispersion* as following:

$$R(A_i, C_j) = \begin{cases} 1 & \psi_{ij} > \frac{\alpha}{m}, \text{(Assume } 1 \leq \alpha \leq m) \\ 0 & \text{otherwise} \end{cases}$$

With these simplifications, we can define the dispersion measure between the clustering result $\mathcal{C}$ and a given class

A_i as:

$$\psi_{\mathcal{C}}(A_i) = \begin{cases} 1 - \frac{\sum_{j=1}^{j=m} R(A_i,C_j)-1}{m} & \text{if } \sum_{j=1}^{j=m} R(A_i,C_j) \neq 0 \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

And finally, the dispersion between the classification result and the clustering one can be defined as follow:

$$\psi_{\mathcal{C}}(A_1, \cdots A_n) = \frac{1}{n}\sum_{i=1}^{n} \psi_{\mathcal{C}}(A_i) \quad (2)$$

Dispersion on the Running Example. In Table 1, the *dispersion* score of each given concept w.r.t. the clustering results is computed using equation 1 and $\alpha = 1$. Based on the table, concepts still and bike have a dispersion score less than 1, because their instances are distributed widely across the different clusters than it is the case for the other concepts. The lower the dispersion scores are, the more difficult to handle for the next level of the hierarchy. According to equation 2, the total dispersion is $\psi_{\mathcal{C}}(a_1, \ldots, a_8) = 0.937$.

Definition 2 (Cohesion ϕ). *The cohesion of classes $A_1, \ldots, A_n$ w.r.t. to a given clustering $\mathcal{C}$ measures the co-appearance of classes together in each cluster.*

This measure satisfies the following conditions for the worst and best cases:

$$\phi_{\mathcal{C}}(A_1, .., A_n) =$$

$$\begin{cases} 1 & \text{if } \forall c, d, i \in \{1..k\} \ |\psi_{ic} - \psi_{id}| = \pm\epsilon \\ 0 & \text{if } \forall c, d \in \{1..k\}, \exists i \in \{1..m\} \ |\psi_{ic} - \psi_{id}| = 1 \pm \epsilon \end{cases}$$

where ϵ is a given small value. From statistical point of view, there exist several possibilities to compute the *dispersion* measure. The following one gives the empirical and more simplest one. For a given cluster $C_l \in \mathcal{C}$ (where $|\mathcal{C}| = m$), the cohesion of two classes is computed as: $\phi_{C_l}(A_i, A_j) = \dfrac{\min(\psi_{il}, \psi_{jl})}{\max(\psi_{il}, \psi_{jl})}$. Accordingly, the simplest cohesion expression between two given classes can be written:

$$\phi_{\mathcal{C}}(A_i, A_j) = \frac{\sum_{l=1}^{l=m} \phi_{C_l}(A_i, A_j)}{m} \quad (3)$$

And finally, the *cohesion* of a given set of concepts as $\{A_1, \ldots, A_i\}$ w.r.t. a cluster $C \in \mathcal{C}$ and clustering $\mathcal{C}$ (where $|\mathcal{C}| = m$), are computed as following respectively:

$$\phi_{C_j}(A_1, \ldots, A_i) = \frac{1}{i(i-1)}\sum_{k=1}^{i-1}\sum_{l=k+1}^{i} \phi_{C_j}(A_k, A_l)$$

$$\phi_{\mathcal{C}}(A_1, .., A_i) = \frac{1}{m}\sum_{l=1}^{l=m}\frac{1}{i(i-1)}\sum_{k=1}^{i-1}\sum_{j=k+1}^{i} \phi_{C_l}(A_k, A_j) \quad (4)$$

Cohesion on the running example. The pairwise cohesion scores for the given clustering example $\phi_{\mathcal{C}}(a_i, a_j)$, can be computed using equation 3 as in Table 1. The cohesion score of all concepts w.r.t. the clusters, i.e. $\phi_{C_i}(a_1, \ldots, a_8)$ is computed in Table 1. The table 1 also shows that for this application it is interesting to learn the following concepts together: still and bike (Clusters C_1 and C_3), walk and run (C_2), and car, bus, train, and subway (C_4). This corresponds to clear semantic biases learned during clustering step and not explicitly introduced.

Algorithm 1: computeHierarchy

```
Input : (i) $\mathcal{E} = \{(X_i, A_i)\}_{i=1}^{|\mathcal{E}|}$ set of annotated
        training examples; (ii) $\mathcal{A} = \{A_1, \ldots, A_n\}$
        denotes the set of concepts; (iii) Distance
        measure $D$ to compute the linkage
$\mathcal{D} \leftarrow \{\,\}$ ;                          % set of clustering results
for $t \in 2, \ldots, |\mathcal{A}| - 1$ do
    $\mathcal{C} = \{C^{(1)}, \ldots, C^{(t)}\} \leftarrow$ cluster($\mathcal{X}$, $D$)
    Compute dispersion $\psi_{\mathcal{C}}(\mathcal{A})$ ;        % using Eqn. 2
    Compute cohesion $\phi_{\mathcal{C}}(\mathcal{A})$ ;          % using Eqn. 4
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathcal{C}, \psi_{\mathcal{C}}, \phi_{\mathcal{C}})\}$
end
$\mathcal{C}^* \leftarrow$ bestClustering($\mathcal{D}$) ;                % using Eqn. 5
foreach $C \in \mathcal{C}^*$ do
    $\mathcal{A} \leftarrow$ getClasses($C$)
    $\mathcal{X} \leftarrow$ getData($\mathcal{A}$)
    if $|\mathcal{A}| = 1$ then
        $Child_i \leftarrow \mathcal{A}$ ; % $i^{th}$ child of the current node
    else
        $Child_i \leftarrow$ computeHierarchy($\mathcal{X}, \mathcal{A}$)
    end
end
Result: Hierarchy $\mathcal{T}$
```

Hierarchy Derivation and Optimization

Thanks to the two proposed measures, it is no longer necessary to enumerate and evaluate all the possible groupings of the search space. This task is delegated to the clustering problem. Therefore, the problem can be reformulated as the search for the best clustering that generates the best grouping of classes. Algorithm 1 describes the recursive process of hierarchy construction from the set of concepts and annotated training examples. It proceeds recursively: given the set of annotated examples $\mathcal{X}$ and the set of concepts $\mathcal{A}$ considered at a given node of the hierarchy (starting from the root), the algorithm computes different clustering solutions for a varying number of clusters (from 2 to $|\mathcal{A}| - 1$, the two other extremes being obviously useless). To select the best clustering solution, a natural optimization model based on the two proposed measures can be stated as:

$$\max_{\mathcal{C}} \ \gamma_1 * \psi_{\mathcal{C}}(A_1, \ldots, A_n) + \gamma_2 * \phi_{\mathcal{C}}(A_1, \ldots, A_n) \quad (5)$$

where γ_1 and γ_2 are additional parameters controlling the trade-off between dispersion and cohesion. This optimization model depends on the selected clustering method and its related distance measure.

Leveraging the Hierarchy for Efficient Training

The non-leaf nodes of the derived hierarchy are assigned with learners trained to discriminate between the concepts or groups of concepts found within their descendant leaves. This implies a bi-level optimization problem with $\mathcal{C}$ (the clustering solution at each step) and w (the weights assigned to the non-leaf nodes of the derived hierarchy) as the inner optimization problem. Evaluating the learners' weights

Algorithm 2: Hierarchy training

```
Input : (i) $\mathcal{E} = \{(X_i, A_i)\}_{i=1}^{|\mathcal{E}|}$ set of annotated
        training examples
$\mathcal{T} \leftarrow$ computeHierarchy($\mathcal{X}, \mathcal{A}$) ;        % Algorithm 1
$\mathcal{T}_{\theta_1,\ldots,\theta_t} \leftarrow$ initialize() ;   % Initialize the weights of
    the learners assigned to the hierarchy
while not done do
    foreach $\theta_t$ do
        Let the super-scripted concepts, $A^{(t)} \in \mathcal{A}^{(t)}$,
          be those grouped in node $t$
        Sample mini-batch from $\{(X_i, A_i^{(t)})\}_{i=1}^{n_t}$
        Evaluate $\nabla_{\theta_t}\ell(\theta_t)$ with respect to the
          mini-batch
        Compute adapted parameters with gradient
          descent: $\theta_t' = \theta_t - \eta\nabla_{\theta_t}\ell(\theta_t)$
        Select examples with high(low) entropy for
          the descendants ;                    % See suppl.
    end
end
Result: Trained hierarchy $\mathcal{T}_{\theta_1^*,\ldots,\theta_t^*}$
```

exactly can be prohibitive due to the expensive inner optimization. Here we propose a simple approximation scheme. We take advantage of the structuring of learners and the inheritance property of inductive biases in hierarchies to effectively drive the learning process by circumscribing the search space for each group of concepts. The idea is to approximate the weights by selecting the most appropriate learning examples to train with the learners of the subsequent levels in the hierarchy, without solving the inner optimization completely until convergence. We investigate for this, two strategies that are designed to improve the learning process, namely, (1) *boosting* strategy: the hard examples are weighted so that the learners located in the descendant nodes focus on them; (2) *student-teacher* strategy (Hinton, Vinyals, and Dean 2015): the easy-to-classify examples are selected for training the subsequent learners. We use an additional parameter (temperature $\in (0, 1)$) which decides how hard or easy is it to classify the examples. Algorithm 2 details the learning process in each node of the derived hierarchy (see § B.2 for full algorithm).

Regarding the class predictions, in classical multi-label classification settings, these can be done in non-leaf nodes (Bi and Kwok 2012; Silla and Freitas 2011). In our case, we use leaf-mandatory classification, i.e., the examples are assigned to an atomic concept (leaf of the hierarchy). Pseudocode describing how predictions are performed given the trained hierarchy can be found in the supplementary material (§ B.3).

Experiments and Results

The empirical evaluation of our approach is organized into three axes: (1) we evaluate the recognition performances of the derived hierarchies; (2) we evaluate the impact of the proposed measures on the derived hierarchies and the separability of the considered concepts; finally, (3) we provide a preliminary assessment of the interplay of inductive biases inside the derived hierarchies via the analysis of the importance of the learners' hyperparameters [2]. All training details, hyperparameters, and their sensitivity analysis can be found in the code repository and supplementary materials (see § C and § D).

Experimental Setup

Representative Related Datasets. We use in our experiments, primarily, the SHL dataset which consists of motion sensor data. It is a highly versatile annotated dataset dedicated to mobility-related human activity recognition. It was recorded over a period of 7 months in 2017 in 8 different modes of transportation in real-life setting in the United Kingdom (*0:Still*, *1:Walk*, *2:Run*, *3:Bike*, *4:Car*, *5:Bus*, *6:Train*, and *7:Subway*). The dataset contains multi-modal data from a body-worn camera and from 4 smartphones, carried simultaneously at typical body locations (*Hand*, *Torso*, *Hips*, and *Bag*). The SHL dataset contains 3000 hours of labeled locomotion data in total making it the most important in the literature. It includes 16 modalities such as accelerometer, gyroscope, magnetometer, linear acceleration, orientation, gravity, ambient pressure, cellular networks, etc. For comparison, we also evaluate our proposed approach on two additional representative datasets, the *USC-HAD* and *HTC-TMD*. More details about these datasets can be found in the supplementary material.

- *USC-HAD* (Zhang and Sawchuk 2012) containing body-motion modalities of 12 daily activities collected from 14 subjects (7 male, 7 female) using MotionNode, a 6-DOF inertial measurement unit, that integrates a 3-axis accelerometer, 3-axis gyroscope, and a 3-axis magnetometer;
- *HTC-TMD* (Yu et al. 2014) containing accelerometer, gyroscope, and magnetometer data all sampled at 30Hz from smartphone built-in sensors in the context of energy footprint reduction;

Baselines. we evaluate the flat classification setting using neural networks which constitute our baseline for the rest of the empirical evaluations. To compare our baseline with the proposed hierarchical model, we make sure to get the same complexity, i.e., comparable number of parameters as the largest hierarchies including the weights of the learners. We also use Bayesian optimization based on Gaussian processes as surrogate models to select the optimal hyperparameters of the baseline model (Snoek, Larochelle, and Adams 2012; Osmani and Hamidi 2019). In addition, we compare our proposed approach with the following closely related baselines from the HAR literature:

- **DeepConvLSTM** (Ordóñez and Roggen 2016): a state-of-the-art HAR model encompassing 4 convolutional layers responsible of extracting features from the sensory inputs and 2 long short-term memory (LSTM) cells used to capture their temporal dependence.

[2] Software package and code to reproduce empirical results is publicly available and can be found at https://github.com/sensor-rich/clustering-based-HL

Model	USC-HAD	HTC-TMD	SHL
DeepConvLSTM	65.8±.0028	68.2±.0016	65.3±.012
DeepSense	67.0±.017	68.5±.0032	66.5±.005
AttnSense	68.5±.04	70.1±.005	68.4±.002
Feature fusion	67.2±.001	69.2±.0074	66.8±.0042
Corr. align.	69.5±.004	70.5±.0026	69.1±.06
Proposed	71.8±.001	74.5±.0017	73.7±.006

Table 2: Recognition performances of various state-of-the-art models on different representative related datasets.

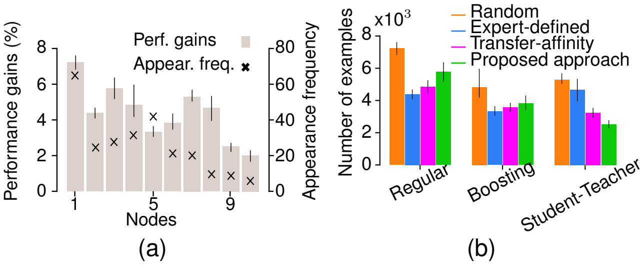

Figure 2: (a) Per-node performance gains, averaged over the entire derived architectures (similar nodes are grouped and their performances are averaged). The appearance frequency of the nodes is also illustrated. (b) The amount of supervision used while training the learners of the hierarchies with the proposed training strategies.

- **DeepSense** (Yao et al. 2017): a variant of the DeepConvLSTM model combining convolutional and a Gated Recurrent Units (GRU) in place of the LSTM cells.
- **AttnSense** (Ma et al. 2019): features an additional attention mechanism on top of the DeepSense model forcing it to capture the most prominent sensory inputs both in the space and time domains and focus on them to make the final predictions.

We use the *meta-segmented* cross-validation (Hammerla and Plötz 2015) for model evaluation to alleviate the problem of *neighborhood bias* and performance over-estimation. Additional details on the evaluation setup and implementation can be found in the supplementary material (§ E).

Performances of the Derived Hierarchies

Table 2 compare the recognition performances obtained with the baseline models on the considered representative datasets. As shown in the table, our proposed approach performs well on the three considered datasets. Note also that performance of the related baselines as reported in the literature confirm the significant issues, analyzed in (Hammerla and Plötz 2015), when using regular cross-validation which are likely leading to overly optimistic performance.

Training the Learners Assigned to the Hierarchy. Figure 2 shows the resulting per-node performances averaged over the entire derived hierarchies, i.e., how accurately the learners assigned to the non-leaf nodes can predict the correct groups of concepts associated to them. Each bar in the figure represents the gained accuracy of each node in our hierarchical approach. For example, the 8th bar corresponds to the concepts 2:*walk*-3:*run*-4:*bike* grouped together. Figure 2b illustrates the amount of supervision on average used at each node of the derived hierarchies using different training strategies (See Algorithm 2). For reference, the amount of supervision required in the regular learning setting is illustrated. The amounts of supervision illustrated in the hierarchical learning settings are those required to attain a comparable accuracy with the regular learning setting. In addition, the amount of supervision is also assessed on (i) randomly picked hierarchies, (ii) the set of domain expert-defined hierarchies, and (iii) hierarchies derived using the approach defined in (Osmani, Hamidi, and Alizadeh 2021), which is based on the transfer-affinity between concepts to build the hierarchies. It is worth noticing that the hierarchies derived using our proposed approach achieve competitive performances while using far less training examples (approx. 2×10^{-3} examples) compared to the other hierarchies. This suggests that the concepts grouping proposed by our measures reflects the actual concepts dependence exhibited in the data. On the other hand, the need for supervision is more pronounced when using the regular training strategy.

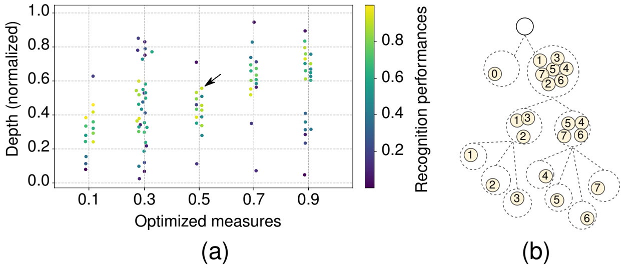

Figure 3: (a) Link between the proposed measures (x-axis) and the properties (depth and arity) of the derived hierarchies (y-axis). The final per-concept recognition performances is also depicted with varying colors. (b) One of the derived hierarchies corresponding to the arrow in the left.

Proposed Measures and Concept Separability

Here we study the correlation between the proposed measures (cohesion and dispersion) and the separability of the grouped concepts. How do the measures of cohesion and dispersion change when we go down the hierarchy? And above all, what is the impact of all this on the derived hierarchies? Are they deeper, i.e., are the best clustering solutions the ones that very quickly decompose the groups of concepts into atomic ones? Or, on the contrary, those which try to keep the concepts grouped until the leaves? How does this affect the learning of groups? How does this ultimately affect the recognition of atomic concepts? Which concepts really benefit from being grouped together? And, which concepts benefit from being rather learned on their own? We assess some of these questions here (see § G for more results).

Figure 3, illustrates the link between the proposed measures and the properties of the derived hierarchies in terms of depth and arity along with the final per-concept recog-

	Groups of concepts			
Hyperparam.	[0][1-7]	[1,2,3][4-7]	[1][2][3]	[4][5][6][7]
First layer				
Kernel size	0.496	0.021	0.026	0.079
# of filters	0.325	0.078	0.014	0.124
Stride	**0.852**	**0.745**	**0.752**	**0.664**
Second layer				
Kernel size	0.147	**0.578**	0.454	0.125
# of filters	0.452	0.327	0.273	0.368
Stride	**0.662**	0.491	**0.765**	0.054
Third layer				
Kernel size	**0.654**	**0.584**	0.027	0.041
# of filters	0.076	0.025	**0.581**	0.031
Stride	0.324	**0.558**	**0.754**	0.017

Table 3: Hyperparameters' importance obtained through the fANOVA analysis of the hierarchy depicted in Figure 3.

nition performances. We particularly focus on the effect of various scores of the cohesion and dispersion measures on the derived hierarchies and what does this imply in terms of concepts grouping and how accurately the atomic concepts are recognized. In theory, optimal hierarchies would be those keeping the concepts grouped while going down the hierarchy, which result in deeper hierarchies in a way that the biases of groups are leveraged to a greater extent. Indeed, this is what we can see for high values of the optimized measures ($\geq$ 0.8), where we get a fairly large number of deep hierarchies which are accompanied by fair recognition performances (approx. 70%). An increase in the computed measures results in a slight augmentation in the recognition performances globally.

Hyperparameters and Inductive Biases

The hierarchical structuring of the concepts allows us to circumscribe the search space for each group of concepts. The bias learned at each non-leaf node is consequently more adapted to each group. However, one question that remains unclear and could open room for further improvement is the link between these various biases. In other words, is there a way to go beyond and structure the biases such that a given learner can share them with its descendent in the hierarchy? Indeed, various works touched this aspect from the operational point of view, such as (Torralba, Murphy, and Freeman 2007; Zhou, Xiao, and Wu 2011) which leveraged transfer of orthogonal representation between children and parents in hierarchies and (Osmani, Hamidi, and Alizadeh 2021) where authors used transfer-affinity between concepts and groups of concepts, but this time to simultaneously build the hierarchy of concepts.

An interesting way to tackle this question is related to the works around weight-agnostic neural architectures and those around the interpretation of the hyperparameters as inductive biases (Lukoševičius and Jaeger 2009; Frankle and Carbin 2018; Gaier and Ha 2019). Here, we provide a solution to investigate the link between the inductive biases used by the learners assigned to one of the derived hierarchies. For this, we design an experimental setting in which the architectures (hyperparameters) of the learners assigned to the non-leaf nodes are optimized in a weight-agnostic fashion. This learning paradigm allows us to shift the focus from the set of weights towards the hyperparameters of the architectures. In a second step, we perform hyperparameter importance assessment following the methodology in (Hutter, Hoos, and Leyton-Brown 2014; Osmani and Hamidi 2018; Hamidi and Osmani 2020) in order to check how inductive biases behave in the learned hierarchy of concepts.

Table 3 summarizes the obtained results from the hyperparameters assessment process. It illustrates the importance of each of the optimized hyperparameters at each node of the considered hierarchy. In particular, among the optimized hyperparameters that define the architecture of the learners assigned to the hierarchy, there are the *kernel size*, *number of filters*, and *stride* of convolution-base neural network layers. Their predefined ranges can be found in the code repository. It is worth noting the appearance, at each level of the hierarchy, of a specific set of hyperparameters that exhibit high importance as captured by the fANOVA framework. In particular, the *stride* of all three layers has the highest importance among this set. This hyperparameter determines the portion of the signal the convolution layers process at a time. The size of this portion is specific to each group of concepts, e.g., smaller for dynamic activities and bigger for static ones.

Conclusion and Perspectives

This paper presents an original approach to deal with the complexity of the hierarchical dependent concepts. The proposed approach starts by clustering groups of atomic concepts close enough to be learned together using cohesion and dispersion measures. The clustering approach reduces substantially the number of tree candidates for grouping the atomic concepts. Empirical evaluations demonstrated superior results using the hierarchies derived using our proposed approach on a dataset collected in real-life settings, which is susceptible to concepts overlaps (in addition to the intrinsic multi-inheritance of the featured concepts). The proposed approach allows us to reduce drastically the exponential theoretical complexity of basic hierarchical learning settings.

Even if the hierarchical structuring of the concepts allows us to circumscribe the search space for each group of concepts and consequently get inductive biases that are more adapted to each group, the proposed model can be further improved to get even better results on the final atomic concepts while using less supervision. As started to be analyzed and discussed in our experiments and explored in some works such as (Torralba, Murphy, and Freeman 2007; Zhou, Xiao, and Wu 2011), the inductive biases learned at each node of the hierarchy can be exhibited and leveraged in a way that some aspects will no longer require to be learned again from scratch. Furthermore, model improvement includes also making the whole process trainable in an end-to-end fashion, which involves formulating the clustering and hierarchy derivation steps in a continuous relaxation scheme.

References

Aghajan, H.; and Cavallaro, A. 2009. *Multi-camera networks: principles and applications*. Academic press.

Bi, W.; and Kwok, J. T.-Y. 2012. Mandatory leaf node prediction in hierarchical multilabel classification. *Advances in Neural Information Processing Systems*, 1: 153.

Bulling, A.; Blanke, U.; and Schiele, B. 2014. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3): 1–33.

Cai, L.; and Hofmann, T. 2004. Hierarchical document categorization with support vector machines. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 78–87.

Essaidi, M.; Osmani, A.; and Rouveirol, C. 2015. Learning Dependent-Concepts in ILP: Application to Model-Driven Data Warehouses. In *Latest Advances In Inductive Logic Programming*, 151–172. World Scientific.

Frankle, J.; and Carbin, M. 2018. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations*.

Gaier, A.; and Ha, D. 2019. Weight agnostic neural networks. In *Advances in Neural Information Processing Systems*, 5364–5378.

Gjoreski, H.; Ciliberto, M.; Wang, L.; Ordonez Morales, F. J.; Mekki, S.; Valentin, S.; and Roggen, D. 2018. The University of Sussex-Huawei locomotion and transportation dataset for multimodal analytics with mobile devices. *IEEE Access*.

Hamidi, M.; and Osmani, A. 2020. Data Generation Process Modeling for Activity Recognition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer.

Hamidi, M.; and Osmani, A. 2021. Human Activity Recognition: A Dynamic Inductive Bias Selection Perspective. *Sensors*, 21(21): 7278.

Hamidi, M.; Osmani, A.; and Alizadeh, P. 2020. A Multi-View Architecture for the SHL Challenge. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct*, 317–322.

Hammerla, N. Y.; and Plötz, T. 2015. Let's (not) stick together: pairwise similarity biases cross-validation in activity recognition. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 1041–1051.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hutter, F.; Hoos, H.; and Leyton-Brown, K. 2014. An efficient approach for assessing hyperparameter importance. In *Proceedings of the International Conference on Machine Learning*, 754–762. PMLR.

Kosmopoulos, A.; Partalas, I.; Gaussier, E.; Paliouras, G.; and Androutsopoulos, I. 2015. Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining and Knowledge Discovery*, 29(3): 820–865.

Lukoševičius, M.; and Jaeger, H. 2009. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3): 127–149.

Ma, H.; Li, W.; Zhang, X.; Gao, S.; and Lu, S. 2019. AttnSense: Multi-level Attention Mechanism For Multimodal Human Activity Recognition. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 3109–3115.

Ordóñez, F. J.; and Roggen, D. 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1): 115.

Osmani, A.; and Hamidi, M. 2018. Hybrid and convolutional neural networks for locomotion recognition. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct*, 1531–1540.

Osmani, A.; and Hamidi, M. 2019. Bayesian Optimization of Neural Architectures for Human Activity Recognition. In *Human Activity Sensing*, 171–195. Springer.

Osmani, A.; Hamidi, M.; and Alizadeh, P. 2021. Hierarchical Learning of Dependent Concepts for Human Activity Recognition. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer.

Quinlan, J. R. 1986. Induction of decision trees. *Machine learning*, 1(1): 81–106.

Samie, F.; Bauer, L.; and Henkel, J. 2020. Hierarchical Classification for Constrained IoT Devices: A Case Study on Human Activity Recognition. *IEEE Internet of Things Journal*.

Scheurer, S.; Tedesco, S.; Brown, K. N.; and O'Flynn, B. 2020. Using domain knowledge for interpretable and competitive multi-class human activity recognition. *Sensors*, 20(4): 1208.

Silla, C. N.; and Freitas, A. A. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2): 31–72.

Snoek, J.; Larochelle, H.; and Adams, R. P. 2012. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, 2951–2959.

Torralba, A.; Murphy, K. P.; and Freeman, W. T. 2007. Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5): 854–869.

Wehrmann, J.; Cerri, R.; and Barros, R. 2018. Hierarchical multi-label classification networks. In *International Conference on Machine Learning*, 5075–5084.

Yao, H.; Wei, Y.; Huang, J.; and Li, Z. 2019. Hierarchically Structured Meta-learning. In *Proceedings of the International Conference on Machine Learning*, 7045–7054.

Yao, S.; Hu, S.; Zhao, Y.; Zhang, A.; and Abdelzaher, T. 2017. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the International Conference on World Wide Web*, 351–360.

Yu, M.-C.; Yu, T.; Wang, S.-C.; Lin, C.-J.; and Chang, E. Y. 2014. Big data small footprint: the design of a low-power classifier for detecting transportation modes. *Proceedings of the VLDB Endowment*, 7(13): 1429–1440.

Zhang, M.; and Sawchuk, A. A. 2012. USC-HAD: a daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 1036–1043.

Zhou, D.; Xiao, L.; and Wu, M. 2011. Hierarchical classification via orthogonal transfer. In *Proceedings of the International Conference on Machine Learning*, 801–808.