



Towards Hierarchical Classification of Data Streams

Antonio Rafael Sabino Parmezan^(✉), Vinicius M. A. Souza,
and Gustavo E. A. P. A. Batista

Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo,
São Carlos, SP, Brazil
parmezan@usp.br, {vmasouza, gbatista}@icmc.usp.br

Abstract. In data stream mining, state-of-the-art machine learning algorithms for the classification task associate each event with a class belonging to a finite, devoid of structural dependencies and usually small, set of classes. However, there are more complex dynamic problems where the classes we want to predict make up a hierarchical structure. In this paper, we propose an incremental method for hierarchical classification of data streams. We experimentally show that our stream hierarchical classifier present advantages to the traditional online setting in three real-world problems related to entomology, ichthyology, and audio processing.

Keywords: Hierarchical classification · Data streams · Online learning

1 Introduction

Most problems in data mining involve the prominent classification task. The purpose of this task is to find, from a significant number of examples, a function that maps the features that describe each example in its respective known class (category). Besides establishing the relationships between features and a category, the discovery function can predict the class of new examples [1].

Traditional supervised machine learning algorithms lead to the data categorization in a flat way, *i.e.*, they seek to associate each example with a class belonging to a finite, devoid of structural dependencies and usually small, set of classes. However, there are a significant number of problems whose classes can be divided into subclasses or grouped into superclasses. This structural dependence between classes characterizes the hierarchical classification [13].

In the hierarchical classification, supervised machine learning methods induce a hierarchical decision model (hierarchical classifier). Such a model links the

Supported by CNPq [grants #140159/2017-7, and #306631/2016-4]; and FAPESP [grants #16/04986-6, and #18/05859-3]. This material is based upon work supported by the United States Agency for International Development under Grant No AID-OAA-F-16-00072.

features of the examples to a class hierarchy, usually structured as a tree or as an acyclic directed graph, with different levels of specificity and generality. The main advantage of a hierarchical classifier is that it divides the original problem into levels to reduce the complexity of the classification function and give flexibility to the process.

Current hierarchical classification algorithms work in a batch setting, *i.e.*, they assume as input a fixed-size dataset that we can fully store in memory. At this point, a challenge not yet addressed by the data mining community focuses on the proposition, or even adaptation, of hierarchical classification techniques capable of dealing with unlimited and evolving data that arrives over time called data streams [16]. Data streams require real-time responses, adaptive models, and impose memory restrictions.

In this paper, we extend the state-of-the-art proposing the first incremental method of hierarchical classification for the data stream scenario. The algorithm represents the class hierarchy as a tree and performs single path predictions. Our study has three major contributions:

- We design an incremental method based on k -Nearest Neighbors (k NN) [4] for the hierarchical classification of data streams. Our algorithm uses a fixed-size memory buffer and builds a one-class local dataset for each node of the class tree, except for the root node, using a set of positive examples that represent the current class. The classification of a new event from the stream is done top-down. Our method stands out for its simplicity in decomposing the feature space of the original problem into subproblems with a smaller number of classes. Thus, every time we go down through the levels of the class tree, the input and output spaces are reduced;
- We build three stream hierarchical datasets and make them available online. Such data are from real-world problems related to entomology, ichthyology, and audio processing;
- We experimentally compare our algorithm with an online k NN flat classifier. In this comparison, we show that in problems where the class labels naturally make up a hierarchy, hierarchical classification methods provide better results than those obtained by flat classifiers. Although other studies have evidenced this fact with static batch learning [3, 13], our work is the first which considers the data stream scenario.

The remaining of this paper is structured as follows: Sect. 2 introduces the fundamentals of hierarchical classification and data streams. Section 3 describes our proposal. Section 4 specifies our experimental evaluation. Section 5 presents results and discussion. Finally, Sect. 6 reports our conclusions.

2 Background and Definitions

We provide in this section the main concepts and definitions of hierarchical classification and data streams, which are essential for understanding our proposal.

Hierarchical Classification. Flat classification differs from hierarchical one by the way domain classes are organized. In flat classification, while a portion of the problems is discerned by the non-existence of interrelationships between classes (single-label classification), the other part is characterized by non-structural relationships between labels (multi-label classification). Structural dependencies, which reflect super or subclass relations, configure hierarchical classification.

A dataset in the attribute-value format for hierarchical classification contains N pairs of examples (\vec{x}_i, Y_i) , where $\vec{x}_i = (x_{i_1}, x_{i_2}, \dots, x_{i_M})$ and $Y_i \subset L = \{L, L.1, L.2, \dots\}$. That is, each example \vec{x}_i is described by M predictive features (attributes) and has a set of labels Y_i for which there are relationships that respect a previously defined hierarchical class structure. The class attribute, in turn, represents the concept to be learned and described by the built hierarchical models using supervised machine learning algorithms.

Hierarchical classification methods can be distinguished according to four main aspects [13]. The first one refers to the type of hierarchical structure – tree or Direct Acyclic Graph (DAG) –, used to represent the relationships between classes. In the tree structure (Fig. 1(a)), each node, except the root node, is associated with at most one parent node. In the DAG structure (Fig. 1(b)), each node, except the root node, can have one or more parent nodes.

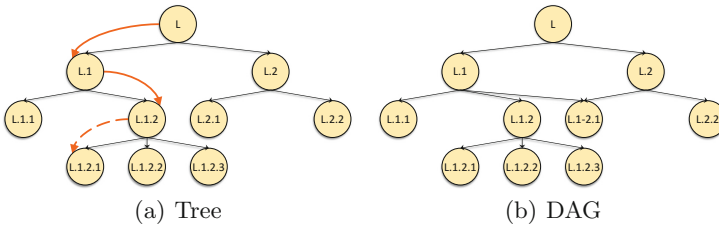


Fig. 1. Hierarchical class structures

The second aspect indicates whether the algorithm can predict classes in only one or several (more than one) paths in the hierarchical structure. For example, in the class hierarchy tree of Fig. 1(a), if the model can predict both class L.2.1 and L.2.2 for a given example, which equates to paths $L \rightarrow L.2 \rightarrow L.2.1$ and $L \rightarrow L.2 \rightarrow L.2.2$, then it is able to predict multiple paths – Multiple Path Prediction (MPP). In contrast, when this type of association is not valid, the method performs the Single Path Prediction (SPP).

The third aspect concerns the hierarchical level of the classification. An algorithm can make predictions using only classes depicted by leaf nodes – Mandatory Leaf-Node Prediction (MLNP) – or by using classes represented by any node, internal or leaf, of the hierarchical structure – Non-Mandatory Leaf-Node Prediction (NMLNP). Figure 1(a) illustrates the difference between these strategies. In this figure, the path $L \rightarrow L.1 \rightarrow L.1.2$ represents the NMLNP strategy, while the path $L \rightarrow L.1 \rightarrow L.1.2 \rightarrow L.1.2.1$ portrays the MLNP strategy. We need to

emphasize that the NMLNP is especially useful in applications that opt for the freedom to conduct a more generic prediction, but with higher reliability.

The fourth and last aspect is related to the mode adopted by machine learning methods to manipulate the hierarchical structure. We can divide the approaches described in the literature into three broad groups: (i) flat approach, (ii) local approach, and (iii) global approach. Further details are available in [13].

Data Stream Mining. A data stream is a sequence of examples (or events) generated continuously over time that imposes severe processing and storage limitations [12]. The features that describe such events can undergo variations over time due to the volatility of the dynamic environment in which they are. These changes are known as concept drift [8, 16]. Concept drifts impose the use of adaptive models to maintain a stable classification accuracy over time.

3 Proposed Method

We propose an incremental algorithm based on k NN to classify data streams hierarchically. Our method represents the class hierarchy as a tree and performs single path predictions using a top-down strategy. Algorithm 1 shows the pseudo-code of the proposal.

Algorithm 1. k NN for Hierarchical Classification of Data Streams

```

/* Initial training set  $T$ ; Unlabeled data stream  $DS$ ; Number of nearest neighbors  $k$ ; Reliability
   threshold  $\delta$ ; Size of the memory buffer  $\eta$ ; Set of predicted labels  $\hat{Y}$  for each  $\bar{x} \in DS$  */
Input:  $T, DS, k, \delta, \eta$ 
Output:  $\hat{Y}$ 
1 begin
2    $\mathcal{H} \leftarrow buildClassHierarchy(T)$ ;
3    $\mathcal{H} \leftarrow oneClassLocalDatasetPerNode(\mathcal{H}, T)$ ;
4   for all the  $\bar{x} \in DS$  do
5      $node \leftarrow \mathcal{H}$ ;
6      $\hat{Y} \leftarrow \{node.label\}$ ;
7     while  $\neg node.isLeaf$  do
8        $children \leftarrow node.children$ ;
9        $d \leftarrow \{\}$ ;
10      for  $i \leftarrow 1$  to  $|children|$  do
11         $local\_dataset \leftarrow node[children[i].label]$ ;
12         $d \leftarrow \{d, averageDistance(\bar{x}, local\_dataset, k)\}$ ;
13      end
14       $dists \leftarrow d / \max(d)$ ;
15       $dists \leftarrow dists - \min(dists)$ ;
16      if  $|dists| \leq \delta$  then
17        break;
18      end
19       $predicted\_node \leftarrow children[getIndex(d == \min(d))]$ ;
20       $\hat{Y} \leftarrow \{\hat{Y}, predicted\_node.label\}$ ;
21       $node \leftarrow predicted\_node$ ;
22    end
23     $Y \leftarrow getCorrectLabelSet(\bar{x})$ ;
24     $\mathcal{H} \leftarrow includeExampleLabeled(\mathcal{H}, \{\bar{x}, Y\}, \eta)$ ;
25  end
26 end

```

In the 2nd line of Algorithm 1, we build a class hierarchy \mathcal{H} from the initial labeled training set \mathcal{T} . In the 3rd line, each node of the class hierarchy, except the root node, is associated with a one-class local dataset. We generate these one-class local datasets using the inclusive heuristic admitting only positive examples [2]. In the 7th line, we employ the top-down prediction strategy, so

that to classify an example \vec{x} belonging to stream \mathcal{DS} the hierarchical structure \mathcal{H} is traversed from the root node to an internal or leaf node depending on the value of δ ($\delta \in [0, 1]$). We expanded a node when the average distance of the k examples from the local dataset nearest to \vec{x} is the lowest compared to the others in the same branch and level. In the 16th line, if δ is greater than zero, we apply the NMLNP. In practical terms, we interrupt the traverse in the class hierarchy when the differences between the average distances calculated for each node of the same branch and level are less than or equal to δ . In the 24th line, we associated the current \vec{x} to its set of correct labels and inserted it into the training dataset. Here, we use a fixed-size memory buffer that holds only the most recent η examples of each class.

4 Experimental Setup

We conducted our experiments using three stream hierarchical datasets related to entomology, ichthyology, and audio processing. These datasets are available online at <https://goo.gl/4pxeWx> and are an original contribution of this paper.

We partitioned each dataset into two sets: (i) initial training set, which covers five labeled examples per class. The class is a set of labels that represents a complete path – from the root to a leaf node – in the label hierarchy; and (ii) test set, composed of the remaining unlabeled examples of the dataset.

We apply our method considering Euclidean distance, $k = 3$, $\delta = 0$, and $\eta = 1000$. The parameter δ set to zero indicates a mandatory leaf-node prediction.

We face the proposed algorithm with an online 3NN flat classifier with Euclidean distance and $\eta = 1000$, where we consider the online flat model as a baseline method. To make a fair comparison, we retrieve from the hierarchical structure all the ancestor labels of each class predicted by the flat classifier (complete path). Thus, we can directly compare a label path predicted by the hierarchical algorithm with a label path predicted by the flat method.

We analyzed the results according to the following performance measures [9]: hierarchical Precision (hP), hierarchical Recall (hR), and hierarchical F-measure (hF). They are defined as follows:

$$\text{hP} = \frac{\sum_i |Y_i \cap \hat{Y}_i|}{\sum_i |\hat{Y}_i|} \quad (1) \quad \text{hR} = \frac{\sum_i |Y_i \cap \hat{Y}_i|}{\sum_i |Y_i|} \quad (2) \quad \text{hF} = \frac{(\beta^2 + 1) \times \text{hP} \times \text{hR}}{\beta^2 \times \text{hP} + \text{hR}} \quad (3)$$

In Eqs. 1 and 2, \hat{Y}_i refers to the set of labels predicted for a test example i and Y_i denotes the set of true classes of this example. During the computation of hP and hR, we need to disregard the root node of the label hierarchy, since by definition it is common to all examples.

In Eq. 3, β belongs to $[0, \infty)$ and corresponds to the importance assigned to the hP and hR values. When $\beta = 1$, the two measures have the same weight in the calculation of the final average. With $\beta = 2$, hR receives double the weight given to hP, whereas for $\beta = 1/2$ the inverse situation occurs, *i.e.*, hP receives twice the weight than hR. We use $\beta = 1$.

5 Experimental Results

In this section, we present and discuss the experimental results for each of the three datasets evaluated. Note that the sequence of our discussion accompanies the classification task complexity, which is proportional to the number of classes.

5.1 Online Hierarchical Classification of Insect Species

The first problem evaluated comprises an entomology and public health application of insect species recognition by optical sensors [14]. This task is the core of intelligent traps developed to catch only target species such as disease vectors. The hierarchical dataset built for this problem has 21,722 examples distributed in 14 different classes. To perform the classification, we extracted 33 features from the signals generated by the sensor such as the energy sum of frequency peaks and harmonic positions. Figure 2 illustrates the class hierarchy of this dataset.

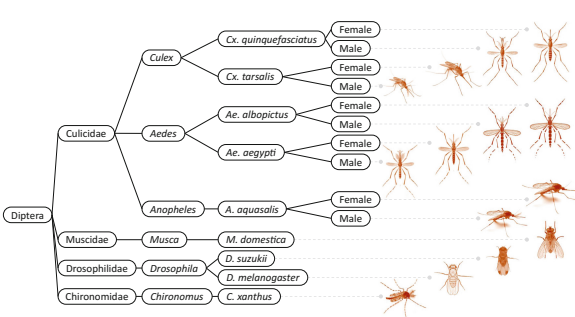


Fig. 2. Hierarchy of insects

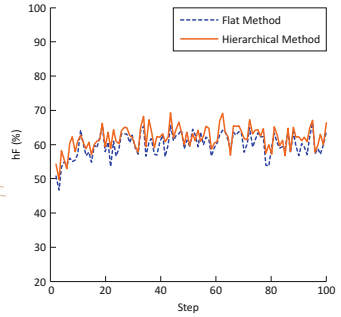


Fig. 3. hF over time for the insects dataset

Figure 3 shows, in terms of hF, the results achieved by the online classification algorithms using the insect species dataset. In this figure, the hierarchical classifier reached a hF of 61.91% while the baseline method obtained a hF of 59.95%. Our algorithm had an average gain of 1.96% on the flat classifier. It is important to note that this difference is small because we “force” our classifier to return the complete paths for the sake of a fair comparison. However, our approach can return less specific outputs in situations with high uncertainty. We deem that a correct generic response is more useful than an incorrect specific one.

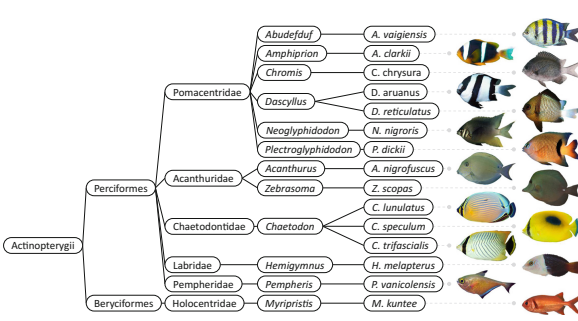


Fig. 4. Hierarchy of fish

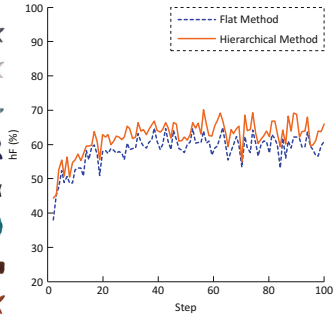


Fig. 5. hF over time for the fish dataset

5.2 Online Hierarchical Classification of Fish Species

Our second hierarchical classification problem contemplates fish species automatic identification based on image analysis [7]. In this problem, we extracted 15 features: 14 based on Haralick descriptors [6] and one involving fractal dimension [10]. Our dataset includes 22,444 examples and 15 classes (Fig. 4).

Figure 5 exhibits the hF results obtained from the online methods employing the fish species dataset. Precisely, the hierarchical model reached a hF of 62.30% while the baseline algorithm achieved a hF of 58.35%. In general, our classifier had a gain of 3.94% over the flat setting.

5.3 Online Hierarchical Classification of Musical Instruments

The last real-world problem evaluated is related to the musical instrument classification based on the analysis of audio signals. We extracted from the signals, 30 features from the Mel Frequency Cepstral Coefficients [11]. The generated dataset has 9,419 examples distributed into 31 classes (Fig. 6).

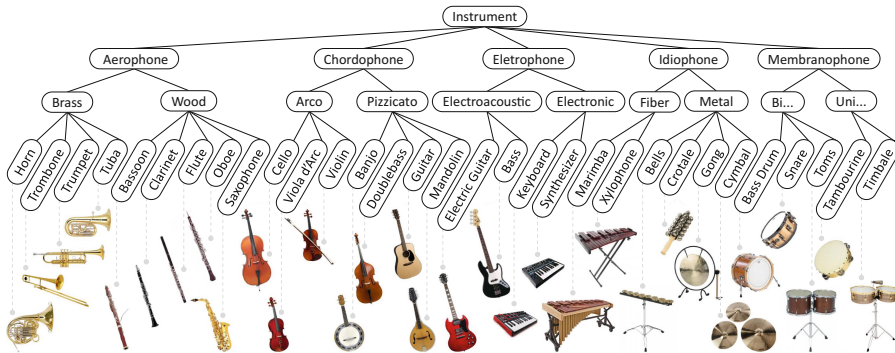


Fig. 6. Hierarchy of musical instruments

Figure 7 displays, in terms of hF, the results achieved from the online classification methods using the musical instruments dataset. In this figure, the hierarchical classifier reached a hF of 84.77% while the baseline model obtained a hF of 82.22%. Our algorithm had an average gain of 2.55% on the flat classifier.

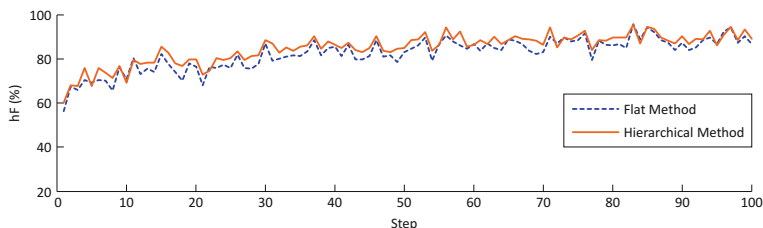


Fig. 7. hF over time for the musical instruments dataset

6 Conclusion and Future Work Prospects

In this paper, we presented an incremental method based on k NN for hierarchical classification of data streams. The algorithm expresses the class hierarchy as a tree and applies a top-down strategy to perform single path predictions. It also adopts a fixed-size memory buffer to store the most recent data and to adapt to concept drift.

To the best of our knowledge, our method is the first that benefits from local information to label events of a stream. We get that local information by decomposing the feature space of the problem into subproblems with a smaller number of classes. Thus, the process gains simplicity, flexibility, and robustness.

We have compared our method with an online k NN flat classifier. The proposed method outperformed the baseline algorithm in all datasets evaluated. This result shows that we can use the information intrinsic to the class hierarchy structure to improve the classification task. In future work, we intend to explore the use of drift detectors [5] and scenarios with delayed labels [15].

References

1. Caruana, R., Karampatziakis, N., Yessenalina, A.: An empirical evaluation of supervised learning in high dimensions. In: ICML, pp. 96–103 (2008)
2. Eisner, R., Poulin, B., Szafron, D., Lu, P., Greiner, R.: Improving protein function prediction using the hierarchical structure of the gene ontology. In: CIBCB, pp. 1–10 (2005)
3. Fagni, T., Sebastiani, F.: On the selection of negative examples for hierarchical text categorization. In: Language Technology Conference, pp. 24–28 (2007)

4. Fix, E., Hodges, J.L.: Discriminatory analysis, nonparametric discrimination, consistency properties. Technical report 4, Project 21–49-004, US Air Force School of Aerospace Medicine (1951)
5. Gonçalves Jr., P.M., Santos, S.G.T.C., Barros, R.S.M., Vieira, D.C.L.: A comparative study on concept drift detectors. *Expert Syst. Appl.* **41**(18), 8144–8156 (2014)
6. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. *IEEE SMC* **3**(6), 610–621 (1973)
7. Joly, A., et al.: LifeCLEF 2016: multimedia life species identification challenges. In: Fuhr, N., et al. (eds.) CLEF 2016. LNCS, vol. 9822, pp. 286–310. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44564-9_26
8. Kelly, M.G., Hand, D.J., Adams, N.M.: The impact of changing populations on classifier performance. In: SIGKDD, pp. 367–371 (1999)
9. Kiritchenko, S., Matwin, S., Famili, A.F.: Functional annotation of genes using hierarchical text categorization. In: ACL-ISMB Workshop (2005)
10. Lee, H.D., et al.: Dermoscopic assisted diagnosis in melanoma: reviewing results, optimizing methodologies and quantifying empirical guidelines. In: KBS (2018)
11. Logan, B.: Mel frequency cepstral coefficients for music modeling. *ISMIR* **270**, 1–11 (2000)
12. Nguyen, H.L., Woon, Y.K., Ng, W.K.: A survey on data stream clustering and classification. *KAIS* **45**(3), 535–569 (2015)
13. Silla, C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. *DMKD* **22**(1), 31–72 (2011)
14. Souza, V.M.A., Silva, D.F., Batista, G.E.A.P.A.: Classification of data streams applied to insect recognition: initial results. In: BRACIS, pp. 76–81 (2013)
15. Souza, V.M.A., Silva, D.F., Gama, J., Batista, G.E.A.P.A.: Data stream classification guided by clustering on nonstationary environments and extreme verification latency. In: SDM, pp. 873–881 (2015)
16. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. *Mach. Learn.* **23**(1), 69–101 (1996)