International Journal of Approximate Reasoning 68 (2016) 179–193

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Hierarchical multilabel classification based on path evaluation ☆

Mallinali Ramírez-Corona, L. Enrique Sucar, Eduardo F. Morales *

Instituto Nacional de Astrofísica, Óptica y Electrónica, Luis Enrique Erro No. 1, Sta. Ma. Tonantzintla, Puebla, 72840, Mexico

ARTICLE INFO

Article history:
Received 16 December 2014
Received in revised form 19 July 2015
Accepted 21 July 2015
Available online 29 July 2015

Keywords:
Multi-label classification
Hierarchical classification
Chain classifiers

ABSTRACT

Multi-label classification assigns more than one label for each instance; when the labels are ordered in a predefined structure, the task is called Hierarchical Multi-label Classification (HMC). In HMC there are global and local approaches. Global approaches treat the problem as a whole but tend to explode with large datasets. Local approaches divide the problem into local subproblems, but usually do not exploit the information of the hierarchy. This paper addresses the problem of HMC for both tree and Direct Acyclic Graph (DAG) structures whose labels do not necessarily reach a leaf node. A local classifier per parent node is trained incorporating the prediction of the parent(s) node(s) as an additional attribute to include the relations between classes. In the classification phase, the branches with low probability to occur are pruned, performing non-mandatory leaf node prediction. Our method evaluates each possible path from the root of the hierarchy, taking into account the prediction value and the level of the nodes; selecting the path (or paths in the case of DAGs) with the highest score. We tested our method with 20 datasets with tree and DAG structured hierarchies against a number of state-of-the-art methods. Our method proved to obtain superior results when dealing with deep and populated hierarchies.

1. Introduction

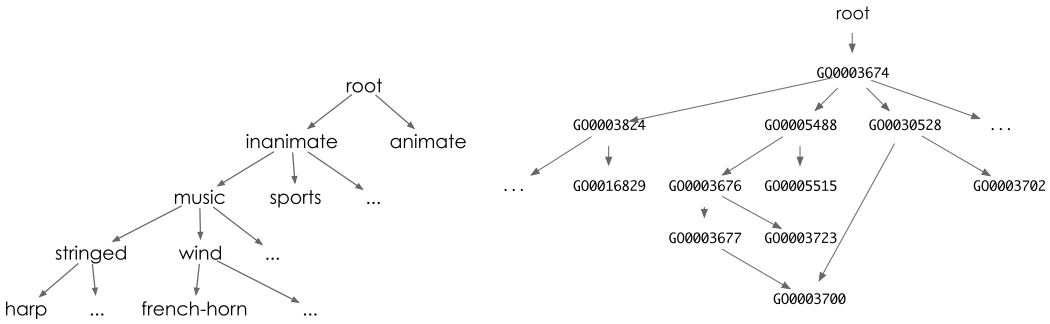
The traditional classification task deals with problems where each example e is associated with a single label $y \in L$, where L is the set of classes, also known as multi-class classification problem. However, some classification problems are more complex and an instance can have multiple labels. When the classes are binary it is called multi-label classification [22,29], when the classes can have multiple class values it is known as multidimensional classification. A multi-label dataset D is composed of N instances, each example has associated a set Y of labels, where $Y \subseteq L$. The task is called Hierarchical Multi-label Classification (HMC) [20] when the labels are ordered in a predefined structure, typically a tree or a Direct Acyclic Graph (DAG); the main difference between them is that in a DAG a node can have more than one parent node (see Fig. 1).

We propose a novel HMC approach, Chained Path Evaluation (CPE), that follows a local classifier approach, by training a local classifier for each non-leaf node in the hierarchy. This decomposition of the problem makes it possible to handle

☆ This is an extended version of the article presented at PGM 2014 [16] that incorporates several additional experiments and a more detailed comparison and analysis. In particular, we include an analysis of the depth effect in the performance of the method, a comparison against the flat approach, and additional experiments with other more complex hierarchies in a different application field.

* Corresponding author.
E-mail addresses: mallinali.ramirez@inaoep.mx (M. Ramírez-Corona), esucar@inaoep.mx (L.E. Sucar), emorales@inaoep.mx (E.F. Morales).

http://dx.doi.org/10.1016/j.ijar.2015.07.008
0888-613X/



(a) Tree structure - Caltech-256 (b) DAG structure - Cellcycle GO

Fig. 1. An example of tree and DAG structured datasets (see Subsection 4.1). The complete hierarchy for the Cellcycle GO data set is shown in Fig. 5 in Appendix A.

large datasets. In contrast to previous local hierarchical approaches, the proposed method has several improvements, that constitute the main contributions of this work. These are listed below and explained in more detail in the paper.

- It adds an extra attribute to the instances in each node with the prediction of its parent node to include the relations between the classes.
- It incorporates a weighting scheme to value more the predictions of the more general classes over more specific ones.
- It scores all the paths, giving a complete overview of the possible predictions, based on the combined probabilities and positions of the nodes in the path for selecting the best one. CPE predicts single paths from the root down to another node (leaf or non-leaf) for tree hierarchies and multiple paths to a single node for DAG hierarchies.
- It obtains predictions where the most specific label is not necessarily a leaf node, what is known as a non-mandatory leaf node prediction (NMLNP), with a novel pruning phase performed before selecting the best path. The pruning discards the branches with less probability to appear in the real label set, thus avoiding potential errors.

We noticed that many evaluation measures score the short paths that only predict the most general classes better than longer and more specific paths, that is why we also propose a new evaluation measure that avoids the bias toward conservative predictions in the case of NMLNP.

The proposed approach was experimentally evaluated with 12 tree and 8 DAG hierarchical datasets in the domain of protein function prediction and image categorization. We concluded that the proposed method, CPE, performs better, than other state-of-the-art methods in deep and populated hierarchies.

The document is organized as follows. Section 2 reviews the relevant work in the area, Section 3 describes the method in detail, Section 4 outlines the experimental setup, in Section 5 the proposed approach is evaluated experimentally and contrasted to other methods, and Section 6 summarizes the paper and suggests possible future work.

2. Related work

When the labels in a multi-label classification problem are ordered in a predefined structure, typically a tree or a DAG, the task is called HMC. The class structure is represented using "IS-A" relations; these relations in the structure are asymmetric (e.g., all *harps* are *stringed* instruments, but not all *stringed* instruments are *harps*) and transitive (e.g., all *harps* are *stringed* instruments, and all *stringed* instruments are *music* instruments; therefore all *harps* are *music* instruments).

In HMC, an example that belongs to a certain class automatically belongs to all its superclasses (hierarchy constraint). When a prediction fulfills the hierarchy constraint it is called a consistent prediction. An example (using the hierarchy in Fig. 1a) of a consistent prediction would be *inanimate, music, stringed, harp*; and an example of an inconsistent prediction would be *inanimate, music, stringed, french-horn*.

There are two kinds of predictions [20]: Mandatory Leaf Node Prediction (MLNP), that returns paths that reach a leaf node in the hierarchy, and Non-Mandatory Leaf Node Prediction (NMLNP) that returns paths that can end in an intermediate node of the hierarchy.

HMC methods are grouped according to the exploration policy they use to solve the classification problem. The most common policies or approaches are: flat, global and top-down.

The flat classification approach predicts only the leaf nodes ignoring completely the information of the hierarchy. Traditional multi-label classification algorithms conform to this approach. However, this very simple approach has the serious disadvantage of having to build a classifier to discriminate among a possibly large number of classes (all the leaf nodes), and does not take advantage of the information provided by the hierarchy.

The global approach learns a single global model for all classes, i.e., it is able to predict each class (node) of the hierarchy. For instance, in a binary tree of depth 3 there are 14 internal and leave nodes, so the global approach needs to build a classifier with 14 classes. This generated model takes into account the class hierarchy as a whole during a single run of the