Due to technological advances, a massive amount of data is produced daily, presenting challenges for application areas where data needs to be labelled by a domain specialist or by expensive procedures, in order to be useful for supervised machine learning purposes. In order to select which data points will provide more information when labelled, one can make use of active learning methods. Active learning (AL) is a subfield of machine learning which addresses methods to build models with fewer, but more representative instances. Even though AL has been vastly studied, it has not been thoroughly investigated in hierarchical multi-label classification (HMC), a learning task where multiple class labels can be assigned to an instance and these labels are hierarchically structured. In this work, we provide a public framework containing baseline and state-of-the-art algorithms suitable for this task. Additionally, we also propose a new algorithm, namely Hierarchical Query-By-Committee (H-QBC), which is validated on datasets from different domains. Our results show that H-QBC is capable of providing superior predictive performance results compared to its competitors, while being computationally efficient and parameter free.

Active Learning for Hierarchical Multi-Label Classification

Felipe Kenji Nakano · Ricardo Cerri · Celine Vens

Received: date / Accepted: date

Abstract Keywords Active Learning · Hierarchical Multi-Label Classification · Predictive Clustering Trees

1 Introduction

Due to recent advances in technology, an exponential amount of data is produced daily, having an impact on many scientific areas. From the machine learning perspective, this availability of data is promising, since it is well-known that models are likely to perform better when learned from more data. In scenarios where data must be labelled by a domain specialist or by expensive procedures, nonetheless, it may present challenges, since labelling requires a substantial financial and time-wise commitment.

Frequently in such scenarios, the amount of labelled data is scarce, whereas unlabelled data is abundant. As a countermeasure, active learning (AL) provides algorithms capable of identifying the most valuable data points to be labelled. AL is a sub-field of machine learning which provides algorithms capable of identifying data points containing valuable information, allowing models to be built with less, but more representative data (Settles, 2010). In this direction, AL has been applied to various tasks such as medical image annotation (Hoi et al., 2006), recommendation systems (Rubens et al., 2011), intrusion detection (Yang et al., 2018) and text categorization (Lewis et al., 2004). Despite such variety, most of the AL literature focus on standard classification and regression tasks.

Felipe Kenji Nakano and Celine Vens
Department of Public Health and Primary Care - KU Leuven Campus KULAK, and
ITEC, imec research group at KU Leuven
Etienne Sabbelaan 53, 8500 Kortrijk, Belgium
E-mail: felipekenji.nakano@kuleuven.be / celine.vens@kuleuven.be

Ricardo Cerri
Department of Computer Science - Federal University of São Carlos
Rodovia Washington Luís, Km 235 - 13565-905 - São Carlos - SP - Brazil
E-mail: cerri@ufscar.br

An even smaller number of studies have developed AL algorithms for hierarchical classification and multi-label classification, which deals with hierarchically organized class labels and with instances that are associated to more than one label, respectively. The intersection of these domains, hierarchical multi-label classification (HMC), has received even less attention.

This lack of studies of AL in HMC is surprising since many applications are addressed by HMC, for instance protein function prediction using underlying taxonomies such as the Gene Ontology[1] (Valentini, 2010; Yu et al., 2017; Nakano et al., 2019; Cerri et al., 2019; Zhao et al., 2020), annotation of non-coding RNA (Zhang et al., 2017), IT services (Zeng et al., 2017) and text annotation (Wang et al., 2011). The development of AL algorithms for HMC will facilitate the development of machine learning models for such tasks, building more efficient models in the expense of less monetary resources. Even other associated tasks, such as data stream (Krawczyk et al., 2017) or time-series classification (Athanasopoulos et al., 2020), could benefit from AL in general since they can present an underlying hierarchical structure.

To the best of our knowledge, the literature presents only a single study in this field (Yan and Huang, 2018). Although obtaining exceptional performance, the authors presented a computationally expensive optimization algorithm validated on adapted HMC datasets with a small number of labels. Moreover, they assume that the oracle decides on the presence of a label given data points by ignoring the ancestors (i.e. more general) labels.

In this work, we have investigated further AL for HMC. We present a simple and efficient algorithm, hierarchical query-by-committee (H-QBC), which is capable of exploiting the underlying hierarchy, while providing superior results, and being more computationally efficient than the method proposed by Yan and Huang (2018). Our experiments also show that H-QBC performs better both on datasets where all labels have the same cost, and increasing costs per depth. For the experiments, we have implemented baseline comparison algorithms, such as uncertainty sampling (Lewis and Catlett, 1994) and query-by-committee (Seung et al., 1992), within the publicly available predictive clustering tree framework for HMC, Clus (Vens et al., 2008). Thereby, allowing reproducible research in this field.

The main contributions of this work are summarized as follows.

- A formal definition of AL for HMC;
- A literature review of AL in HMC, and in its related tasks of non-hierarchical multi-label classification and hierarchical single-label classification;
- A new AL algorithm for HMC, namely H-QBC, which is implemented within the Clus Predictive Clustering Tree framework (Vens et al., 2008) where we also included the other algorithms evaluated in this study[2];

The remainder of this paper is organized as follows. In Section 2, we provide the theoretical background including a description of the hierarchical multi-label classification, the definition of active learning for hierarchical multi-label classification and related work; Further, Section 3 introduces our new algorithm and presents details on how the hierarchy of labels is used; Next, Section 4 describes

[1] http://geneontology.org/

[2] https://itec.kuleuven-kulak.be/supportingmaterial

the datasets and experimental setup, followed by results and discussion (Section 5); Finally, in Section 6 we state our conclusions and directions for future research.

2 Background

In this section, we present the theoretical background of our work. At first, we bring more details about HMC, followed by a definition of AL applied to HMC, and recent work on AL for non hierarchical multi-label classification and hierarchical classification.

2.1 Hierarchical Multi-Label Classification

In traditional classification, given a set of classes Y and a space of instances $\mathbf{X}$, a model f is trained to assign a single class $y_i \in Y$ for every $\mathbf{x}_i \in \mathbf{X}$. In this task, classes are mutually exclusive. However, some real world cases present a pre-defined structure of classes, configuring a hierarchical classification problem. When also multiple classes $y_i \in Y$ can be assigned to every $\mathbf{x}_i \in \mathbf{X}$, we call the task hierarchical multi-label classification (HMC).

Formally, as stated by (Vens et al., 2008), hierarchical multi-label classification problems are defined as follows:

Given:

- A instance space $\mathbf{X}$;
- A class hierarchy $(Y, \leq_h)$, where Y is a set of classes and $\leq_h$ is a partial order, representing the superclass relationship ($\forall y_1, y_2 \in Y : y_1 \leq_h y_2 \iff y_1$ is a superclass of y_2);
- A set of instances T in the format $(\mathbf{x}_i, S_i)$, with $\mathbf{x}_i \in \mathbf{X}$ and $S_i \subseteq Y$;
- A quality criterion q which rewards models with high performance and low complexity.

Find:

- A function $f : \mathbf{X} \rightarrow 2^Y$, 2^Y being the power set of Y, such that f maximizes q and $y \in f(\mathbf{x}) \rightarrow \forall y' \leq_h y : y' \in f(\mathbf{x})$. This last condition is referred to as the hierarchy constraint.

The criterion q changes according to the predictive model used. For instance, in neural networks, such as the one proposed by Wehrmann et al. (2018), q consists of a loss function. Differently from that, in the genetic algorithms proposed by Cerri et al. (2019), it consists of an heuristic. As for decision trees, q is associated to the evaluation function.

The literature offers two HMC approaches: global and local. The global approach builds a single classifier capable of dealing with hierarchical constraints. Algorithms used to build classifiers such as decision trees and genetic algorithms are adapted (Vens et al., 2008; Cerri et al., 2012, 2019). In this approach, predictions are performed in a single step. Even though computationally faster, the global approach requires more complex implementations.

On the other hand, the local approach employs techniques to convert complex problems to well-studied ones. The main classification problem is transformed into many sub-problems, each dealing with the prediction of a subset of labels. For instance, Vens et al. (2008) proposed to train one classifier for every edge of the hierarchy. Alternatively, Nakano et al. (2017) proposed strategies that train a classifier for every local parent node.

2.2 Active Learning

In conventional supervised learning, the model is trained using a static labeled training set. Differently from that, active learning (AL) is a field of machine learning that allows classifiers to be built with less, but more representative data. More specifically, AL addresses applications whose labelled data is very limited, but unlabelled is easily obtainable.

In these cases, building a predictive model using only the labelled data is very challenging, and it often leads to undesirable results. Additionally, obtaining the labels of the entire unlabelled dataset is not feasible since it might require financial and time-wise commitment, and some data points might not be useful for the model. In this direction, AL algorithms investigates methods to select the most informative data points to be labelled. In Figure 1, we describe how AL is employed.

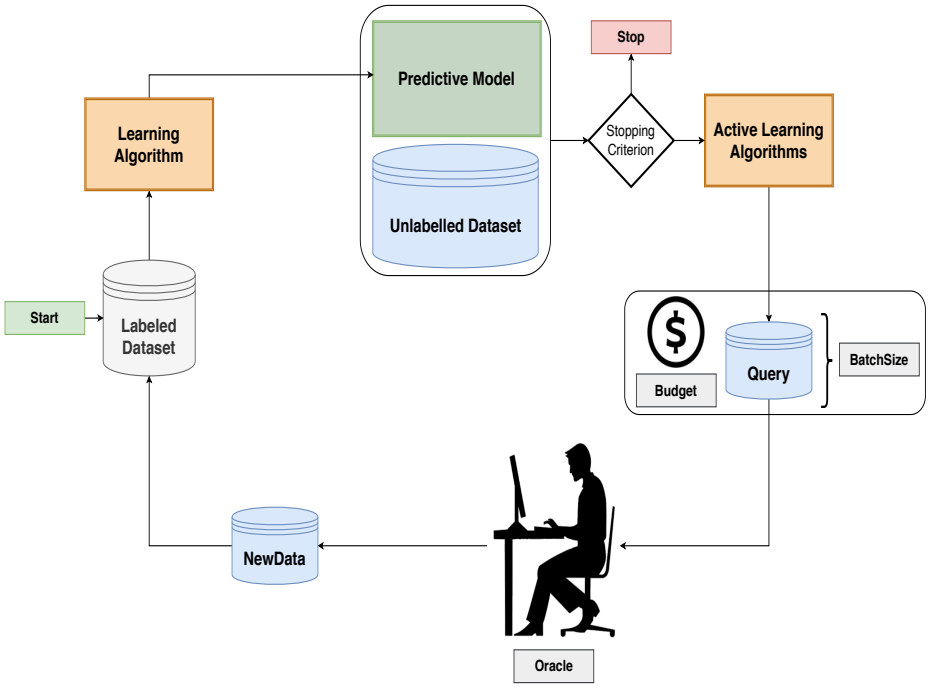


Fig. 1: Example of Active Learning cycle

At first, a predictive model is built using the initial labeled dataset. Further, data points are selected to be labelled. This selection is performed by prioritizing a criterion which represents how meaningful each data point from the unlabelled dataset is. Many different criteria have been proposed, and most of them rely on the uncertainty of the model. For instance, uncertainty sampling, a common baseline algorithm, selects data points which are located closest to the decision boundary of the model. Next, these data points are given to a domain specialist to be labelled.

The domain specialist, more generally denoted as the oracle, consists of a data annotator who provides the ground truth of the selected data points, and these new data points are concatenated to the labelled dataset. The definition of oracle changes according to the application. In medical applications, the oracle is a trained physician (Hoi et al., 2006); in a recommendation system, the oracle may be a user providing feedback on a recommendation (Rubens et al., 2011). Some applications may even adopt the concept of crowdsourcing (Yan et al., 2011).

Next, the predictive model is rebuilt with the new labelled data. This procedure is repeated until a stopping criterion is met. The stopping criterion may also change from application to application. In some cases, new labels are obtained until the model achieves satisfactory performance, or until the budget is depleted.

The field of active learning is closely related to that of semi-supervised learning (SSL). In both cases one tries to make optimal use of a set of unlabeled instances. In AL we obtain the ground truth label via the oracle, whereas semi-supervised methods rely on different assumptions. For instance, self-training methods (van Engelen and Hoos, 2019) follow the smoothness assumption which states that if two points are located closely in the input space, then they should be assigned with the same label or target (Levatić et al., 2017). The notion of reliability is crucial for both domains: whereas SSL focuses on the most reliable predictions, AL prioritizes the least reliable predictions. This notion of reliable predictions can also be used to increase interpretability of models, as performed by Ribeiro et al. (2016) and Štrumbelj and Kononenko (2014).

2.3 Active Learning for Hierarchical Multi-Label Classification

In the following, we present a formal description of the task we consider in this article. We extend the definition of HMC to include an AL component (in one AL cycle).

Given:

- A instance space $\mathbf{X}$;
- A class hierarchy $(Y, \leq_h)$, where Y is a set of classes and $\leq_h$ is a partial order, representing the superclass relationship ($\forall y_1, y_2 \in Y : y_1 \leq_h y_2 \iff y_1$ is a superclass of y_2);
- A set of labelled instances L in the format $(\mathbf{x}_i, S_i)$, with $\mathbf{x}_i \in \mathbf{X}$ and $S_i \subseteq Y$;
- A set of unlabelled instances U in the format $(\mathbf{x}_i, ?)$, with $\mathbf{x}_i \in \mathbf{X}$;
- An Oracle O, capable of labeling data from U;
- A quality criterion q which rewards models with high performance and low complexity;

- A function $f : \mathbf{X} \rightarrow 2^Y$, 2^Y being the power set of Y, such that f maximizes q and $y \in f(\mathbf{x}) \rightarrow \forall y' \leq_h y : y' \in f(\mathbf{x})$;
- A budget per iteration B that limits the overall expenses of O;
- A batch size $Batchsize$ that limits the number of data points to be labelled by O in the current AL cycle.
- A cost vector in the format $[c]^{Levels}$ that defines the cost of labels according to their depth. In this case, $Levels$ stands for the number of levels in the hierarchy of the problem addressed;

Find:

- a subset of query instances ($Query$), composed of pairs (U_q, Y_q), of maximal size $BatchSize$, such that $(U_q, ?) \in U$, $Y_q \in Y$, and f maximally improves when trained on $Query \cup L$ instead of on L, after the Oracle labels all elements of $Query$ (provides values for Y_q). The cost of labelling Q should not exceed the budget available at the iteration.

The query can be built using two types of methods: instance-based or label-based.

Most of the initial work on AL, specially the ones which address non-hierarchical multi-label classification problems, have investigated instance-based methods. As its name suggests, instance-based methods focus on the instance perspective by building queries with pairs such as (U_q, Y) where U_q is an instance from U and Y is the entire label set of the problem.

These methods can present disadvantages (Qi et al., 2008; Wu et al., 2017, 2018), however, since f might not benefit from all labels. Additionally, given that AL is often limited by a budget, and given the considerably large number of labels, it may be inefficient to query all labels. Due to this limitation, recent studies on AL for non-hierarchical multi-label classification have proposed label-based methods (Wu et al., 2017).

Label-based methods seek to build queries with pairs in the format (U_q, y_q) where $(U_q, ?)$ is an instance from U and y_q is a particular label of the problem. In this way, instead of labelling an instance with a set of labels, the oracle O only needs to provide an answer for a particular label, i.e. $y_q \in Y$. Hence, local models can be updated with class specific data, increasing their performance rapidly, and reducing the building cost overall.

Label-based methods are further interesting due to the possibility of using multiple oracles. For instance, a particular oracle might be more suitable for a subset of labels, thus rather than answering all labels, the oracle would focus on its own expertise. For instance, in HMC, different oracles might be responsible for different sub-trees of labels, allowing parallel labelling where the same instance is simultaneously answered by multiple oracles. Finally, label-based methods are further promising due to the possibility of exploiting the hierarchy constraint, which we will address in Section 3.

2.4 Related Work

In Tables 1 and 2, we provide an overview on AL studies for non-hierarchical multi-label classification (MLC) and hierarchical classification, respectively. In what follows, we adopt the acronyms: support vector machine (SVM), logistic regression

(LR) and k-nearest neighbours (KNN) to describe the underlying base classifier used. Mind that, if a study presented more than one algorithm, we are reporting only the best one.

2.4.1 Active Learning for non-hierarchical multi-label classification

	#Measure	#Approach	#Base Classifier
Li et al. (2004)	Uncertainty	Instance-Based	SVM
Brinker (2006)	Uncertainty	Instance-Based	SVM
Yang et al. (2009)	Quantification	Instance-Based	SVM and LR
Hung and Lin (2011)	Hamming Loss	Instance-Based	SVM and LR
Chakraborty et al. (2011)	Uncertainty	Instance-Based	SVM
Li and Guo (2013)	Uncertainty	Instance-Based	SVM
Cherman et al. (2017)	Uncertainty (Deviation)	Instance-Based	SVM
Reyes et al. (2018)	Uncertainty	Instance-Based	SVM
Qi et al. (2008)	Uncertainty	Label-Based	Kernelized Entropy
Zhang (2009)	Uncertainty	Label-Based	SVM
Zhang et al. (2012)	Uncertainty	Label-Based	SVM
Huang and Zhou (2013)	Uncertainty	Label-Based	SVM
Huang et al. (2014)	Uncertainty	Label-Based	SVM
Zhang et al. (2014)	Uncertainty	Label-Based	SVM
Vasisht et al. (2014)	Uncertainty	Label-Based	Bayesian
Wu et al. (2014)	Uncertainty	Label-Based	KNN
Ye et al. (2015b)	Uncertainty	Label-Based	KNN
Ye et al. (2015a)	Uncertainty	Label-Based	KNN
Guo et al. (2017)	Uncertainty	Label-Based	KNN
Wu et al. (2017)	Uncertainty	Label-Based	KNN
Wu et al. (2018)	Uncertainty	Label-Based	KNN

Table 1: Recent work on active learning for non-hierarchical multi-label learning.

As seen in Table 1, most of the work in MLC relies on uncertainty based disagreement measures. In the first work of the field, to the best of our knowledge, a multi-label SVM was used to select instances whose labels were uncertain (Li et al., 2004). Afterwards, most of the improvements were obtained via slight changes to the uncertainty, alongside an extra criterion. For instance, Brinker (2006) proposed to consider the uncertainty of all labels of an instance; Chakraborty et al. (2011) proposed an optimization approach which optimized uncertainty while reducing redundancy; Li and Guo (2013) proposed to combine uncertainty and the distance between the number of labels predicted as positive and the label cardinality of the labelled dataset. Cherman et al. (2017) proposed an uncertainty, namely deviation, that considers the difference between the average prediction probability of positive labels and the most uncertain negative label. Finally, Reyes et al. (2018) proposed to incorporate a rank aggregation of labels which, to the best of our knowledge, is the current state-of-art in this field.

Using a different approach, Yang et al. (2009) proposed to combine SVMs and a LR model to predict the number of positive labels present in unlabelled instances, similar to a quantification learning approach, whereas Hung and Lin (2011) proposed to reduce the hamming loss, a common evaluation measure in MLC, to select the most informative instances.

Following a similar trend on label-based methods, many studies have proposed different extra criteria to be used alongside the uncertainty. Zhang (2009) proposed to select instance-label pairs which are informative for not only the classifier associated to its label, but to all classifiers involved. Zhang et al. (2012) proposed to combine the Kullback-Leibler divergence with association rule mining algorithms to detect pair-wise label relationships, this work was improved by considering correlations between more than two labels in (Zhang et al., 2014). Wu et al. (2014) evaluated how label specific uncertainty is superior to uncertainty of the entire instance. In a follow-up work, a label correlation criterion measured using cosine similarity was employed (Ye et al., 2015a). In (Ye et al., 2015b; Wu et al., 2017), the cosine similarity was replaced by a chi-square estimation. Similarly, in (Guo et al., 2017) results showed that a low-rank matrix optimization can be superior to the chi-square estimation. Lastly, an extra criterion related to sample noise was added in (Wu et al., 2018), which is the current state-of-art in this area.

As for other approaches, Qi et al. (2008) proposed an adaptation of the expectation maximization algorithm to consider partially labelled instances. Jiao et al. (2014) proposed a two-step algorithm where instances are selected at first, and then a KNN-graph is used to select labels. Huang et al. (2014) proposed a systematic view on obtaining representative and informative instance-label pairs using a min-max strategy. Vasisht et al. (2014) proposed an optimization algorithm which minimizes the uncertainty after labelling the selected subset of instance-label pairs.

It is noticeable that, even though label-based methods were reported as superior (Qi et al., 2008), there is still plenty of work on instance-based methods. Additionally, with the exception of few works (Qi et al., 2008; Zhang, 2009), the evaluation of these studies considers only algorithms from the same sub-field. Thus, there is no comparison between the current state-of-art of instance-based and label-based methods.

Furthermore, most of the research is not concerned about optimal parameters for their base classifiers. The majority of the studies employ a simple model, mostly linear SVMs or KNN, with pre-defined parameters.

2.4.2 Active Learning for hierarchical classification

	#Measure	#Approach	#Base Classifier
Cheng et al. (2012)	Embedding	Instance-Based	SVM
Li et al. (2012)	Uncertainty	Label-Based	SVM
Li et al. (2013)	Uncertainty	Label-Based	SVM
Cheng et al. (2014)	Embedding	Instance-Based	SVM
Chakraborty et al. (2015)	Uncertainty	Instance-Based	LR
Mo et al. (2016)	Uncertainty	Instance-Based	LR and Conditional Random Fields
Duin (2017)	Uncertainty	Instance-Based	SVM and LR
Yan and Huang (2018)	Uncertainty	Label-Based	SVM

Table 2: Recent work on active learning for hierarchical classification.

As can be seen in Table 2, the literature offers fewer works in the field of hierarchical classification. Hierarchical classification problems are split into two

categories: hierarchical single-label (HSC) and hierarchical multi-label (HMC). HSC problems allow instances to have only a single path of classes (from the root to a particular class, whereas its multi-label counterpart allow multiples paths of classes. Mind that, all studies reported here investigate HSC problems, with the exception of the work of Yan and Huang (2018).

The majority of studies in HSC focus on the instance-based methods. Using embedding methods, the work of Cheng et al. (2012) proposed a variance based uncertainty method for embedding tree methods. In their second study, (Cheng et al., 2014), the variance based method was replaced by a KNN-graph which is used to promote diversity among instances. Chakraborty et al. (2015) proposed a method, called BatchRank, employing an optimization algorithm which maximizes uncertainty and diversity. Even though BatchRank is not the most recent work, it can be considered the state-of-art.

Differently from that, Mo et al. (2016) proposed to use an adapted version of the BANDIT algorithm to select instances. As an application of this work, Duin (2017) investigated its performance in mitochondrial disease protein datasets.

As for label-based methods, in the work of Li et al. (2012, 2013), methods using "multiple oracles", one for each class, were investigated. Their results pointed out that, by propagating positive and negative labels, superior results are achievable with a significant smaller amount of data. In their second work, the authors enhanced their method by using predictions from their current base classifiers.

Lastly, the single study on AL for HMC was performed by Yan and Huang (2018). The authors proposed a label-based method that maximizes the sum of the uncertainty of a subset of classes from the hierarchy, while minimizing its cost. At first, their method uses K-Nearest Neighbours to predict labels from the unlabelled dataset. If the label is predicted as positive, the algorithm evaluates the ancestors labels, otherwise descendants labels are taken in consideration. Next, it uses the bi-objective evolutionary algorithm POSS to select the best Instance-Label pairs according to both informativeness and the cost (Qian et al., 2015).

3 Proposed Method: Hierarchical Query-By-Committee

Initially, we bring an introduction to query-by-committee methods, containing insights about their main principles. Afterwards, we define our proposed algorithm hierarchical query-by-committee. Finally, we propose how the hierarchy constraint is exploited.

3.1 Query-By-Committee

As reported in the previous section, most of the AL work has focused on uncertainty sampling variations. Despite its popularity, these type of methods may select spurious data points since noise can lead to abnormal behaviour.

As a countermeasure, we propose to use query-by-committee (QBC). QBC is a family of active learning algorithms that trains a committee of classifiers with competing hypotheses and builds queries with respect to a disagreement measure (Seung et al., 1992). Hence, the biases of many classifiers are employed instead of a single one, leading to more robustness and flexibility.

QBC algorithms rely on two fundamental elements: a committee of models with different version spaces and a disagreement measure capable of identifying data points in which the committee disagrees the most.

The first element is usually achieved by employing an ensemble of predictive models, such as a random forest. For the second element, it is very common to employ the variance of the prediction probabilities provided by the ensemble. This is further motivated by some works (Cerri et al., 2019; Vens et al., 2008) that showed that the variance can be successfully incorporated into problems with hierarchical structures. Besides that, the variance is a simple disagreement measure where each member of the committee contributes directly to the construction of the query.

We provide a formulation for the variance in Equation 1 where $P_f(y_j|\mathbf{x}_i)$ stands for the prediction probability output by a model in the ensemble f for label y_j associated to the instance $\mathbf{x}_i$; $\overline{P(y_j|\mathbf{x_i})}$ corresponds to the mean probability of label y_j for all models in the ensemble, and F is the number of models. In this case, QBC selects instance-label pairs with the highest values.

$$Var(\mathbf{x}_i, y_j) = \frac{\sum_f (P_f(y_j|\mathbf{x}_i) - \overline{P(y_j|\mathbf{x}_i)})^2}{F-1} \tag{1}$$

3.2 Hierarchical Query-By-Committee

In order to evaluate the informativeness of a particular instance-label pair, we identify three interrelated characteristics of HMC applications that should be exploited:

- Instances typically contain very few positive labels associated to them;
- Positive labels in deeper levels of the hierarchy are rare, but more informative;
- Positive labels are correlated to each other, as established by the underlying hierarchy and the hierarchy constraint;

Many HMC datasets, specially the benchmarks commonly used, have a high number of possible labels when compared to regular multi-label problems. However, most of the instances have few positive labels associated to them. According to Cerri et al. (2015) most of the datasets have an average of 2 labels per instance per level, and some levels might have more than 100 possible classes, leading to heavy imbalance. This is further aggravated in deeper levels of the hierarchy where labels are very rare (average of 1 label per instance), but more informative as they are more detailed.

To take these characteristics into account, we propose an adapted version of QBC. More specifically, we focus on queries with positive and deep labels, while exploring the relationship among labels as established by the hierarchy. The motivation to focus on positive labels is even more important in the positive unlabeled learning setting, which is often encountered in HMC settings (Bekker and Davis, 2018). In this setting, there are no truly negative labels: a label is negative by absence of evidence of being positive.

Similarly to QBC, we use the variance as the key component to select queries, since, as we show in Figure 2, it allows to identify positive instance-label pairs. Indeed, positive labels present higher values than the negative ones in most of

the cases. However, the values are still low overall and outliers can lead to the inclusion of negative pairs. As a countermeasure, we propose to incorporate the relationship among labels as established by the hierarchy to promote the selection of positive pairs. More specifically, we employ the variance of ascendants and the descendants of a label

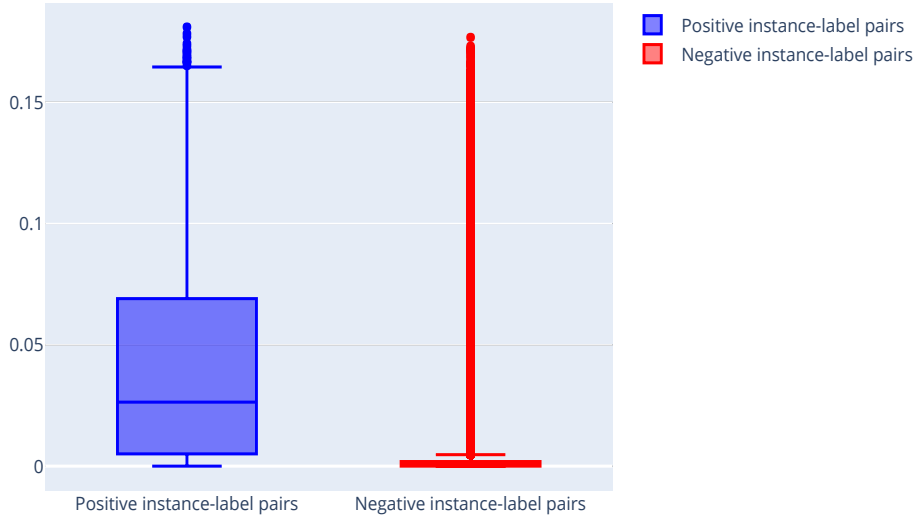


Fig. 2: Boxplot of the variance used in QBC (Equation 1) of the positive and negative label-pairs for the Cellcycle dataset

Incorporating the descendants is related to exploring the sub-tree of candidate query label (with high variance). If a label is truly positive, it is likely that some of its descendants will be positive as well, leading to a higher average variance in the subtree. Hence, we promote the selection of these pairs by including the variance of its descendants. Complementary, we motivate the selection of deeper labels by including the variance of its ascendants. The rationale behind that is that positive and deep labels have a path of positive labels all the way up to the root of the hierarchy, resulting in a large variance for the path.

Considering this, we propose an extended version of query-by-committee, namely Hierarchical Query-By-Committee (H-QBC). We present a formulation for H-QBC in Equation 2 where $Var(\mathbf{x_i}, y_j)$ is defined in Equation 1, $Anc(y_j)$ stands for the ancestors of y_j, $Desc(y_j)$ denotes the descendants of y_j and $|.|$ denotes the size of a set. Intuitively, instance-label pairs with the highest values for H-QBC are prioritized in the query.

$$H\text{-}QBC(\mathbf{x_i}, y_j) = Var(\mathbf{x_i}, y_j) + \frac{\sum_{y_a}^{Anc(y_j)} Var(\mathbf{x_i}, y_a) + \sum_{y_d}^{Desc(y_j)} Var(\mathbf{x_i}, y_d)}{|Anc(y_j)| + |Desc(y_j)|} \tag{2}$$

In Figure 3, we provide an example of how our method works for a given instance. For measuring H-QBC (01/01), we sum its own variance (Var(01/01)) and the mean variance of its ancestors (01) and descendants (01/01/01 and 01/01/03).

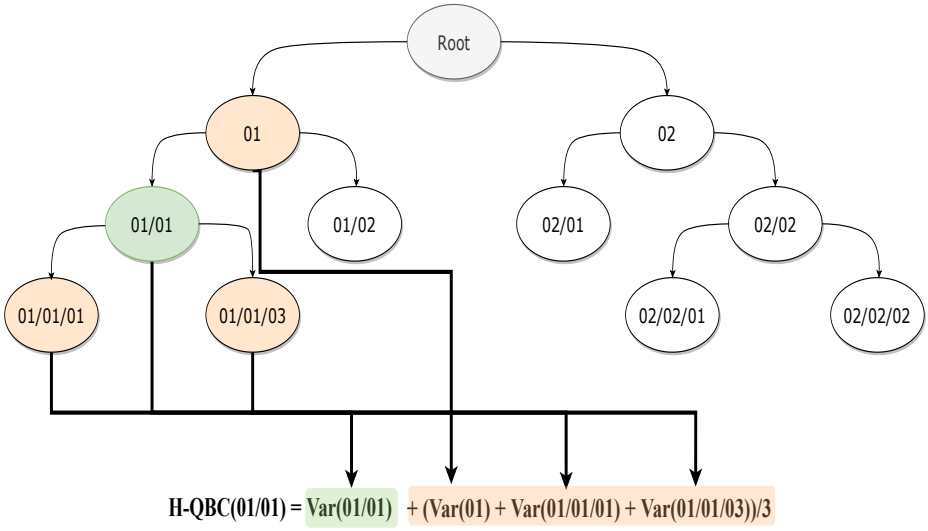


Fig. 3: Example of H-QBC. The value of H-QBC(01/01) corresponds to Var(01/01) + (Var(01) + Var(01/01/01) + Var(01/01/03)) / 3

As seen in Figure 4, the new variance values associated to positive pairs is greater than the regular variance, increasing their chances of being included in the query. There is also a slight increase in the variance for negative pairs, nonetheless by including only the pairs with highest variance, there is a strong tendency to include positives.

H-QBC can present limitations. First, we are focusing explicitly on the correlations between child and parent classes, nonetheless labels from different sub-trees (flat correlations) of the hierarchy may be correlated as well. A simple way to incorporate them would be to include their co-occurrences as proposed by Ye et al. (2015b). On top of that, prioritizing sub-trees of labels might lead to lack of diversity, i.e. the query might contain many labels from the same instance. Such problem can be addressed by employing an extra criterion related to distances between the instances from the unlabeled dataset (Chakraborty et al., 2015). Second, H-QBC demands a committee of classifiers due to the variance. This is not an issue in HMC since complex models are often employed (Wehrmann et al., 2018). Some other tasks, however, such as data streams classification (Krawczyk et al., 2017), might require simpler and faster models, an alternative version might employ a simpler disagreement measure.

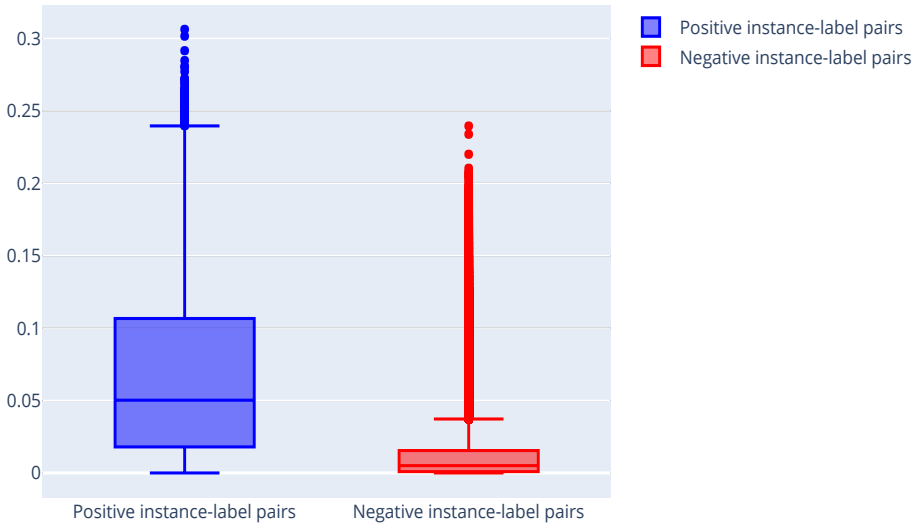


Fig. 4: Boxplot of the variance used in H-QBC (Equation 2) of the positive and negative label-pairs for the Cellcycle dataset

3.3 Exploiting the hierarchy constraint

As previously stated, the hierarchy constraint establishes that $y \in f(\mathbf{x}) \rightarrow \forall y' \leq_h y : y' \in f(\mathbf{x})$. In other words, whenever a data point is labelled with y, all of its ancestors are included as well. This constraint holds for both the labelled training set and predictions produced by f.

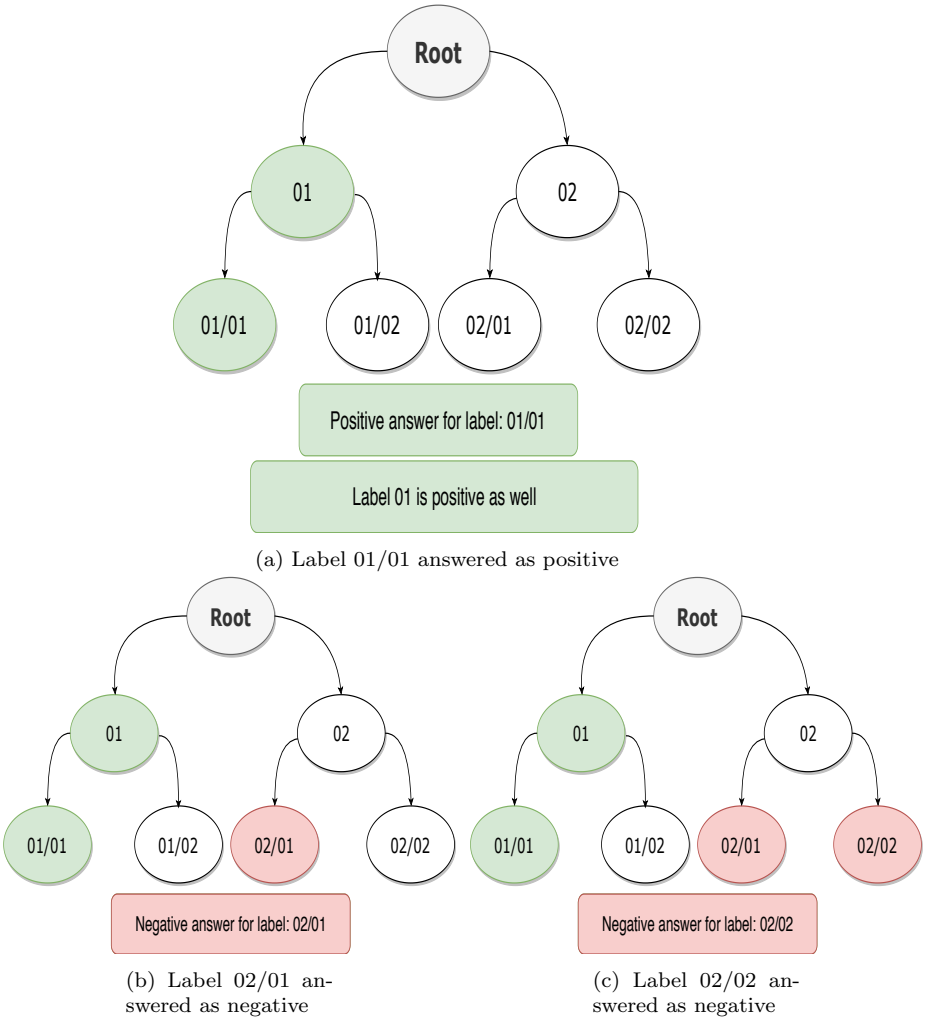
In the current state-of-the-art, hierarchical active learning with cost (HALC) (Yan and Huang, 2018), the constraint is satisfied by querying the instance-label pair requested and propagating its results according to the answer provided by the oracle. In case of a negative answer, all descendants are also negative, and in the opposite case, all ancestors are positive.

In both situations, labels are automatically answered due to the constraint. However, the ancestors of a label are not considered when the Oracle is answering an instance-label pair. Such assumption is not intuitive from a human perspective, and often leads to higher expenses of the budget.

Assuming a query with the following instance-label pairs: $[(U_q, 01/01), (U_q, 02/01), (U_q, 02/02)]$ and cost vector [1,5], and also assuming that the label 01/01 is positive, whereas 02/01 and 02/02 are negative for U_q, the oracle would follow the procedure shown in Figure 5.

Initially, since label 01/01 is positive, the label 01 is automatically labelled as positive as well (Figure 5a). Next, label 02/01 is labelled as negative (Figure 5b). Finally, label 02/2 is also negative (Figure 5c). In this case, the total cost is 5 (01/01) + 5 (02/01) + 5 (02/02), making a total 15.

As a countermeasure, we propose to incorporate all ancestors of the label being queried, respecting the order established by the hierarchy constant. Considering the same query and costs, i.e., $[(U_q, 01/01), (U_q, 02/01), (U_q, 02/02)]$ and [1,5], we adopt the labelling procedure exemplified in Figure 6.



(a) Label 01/01 answered as positive

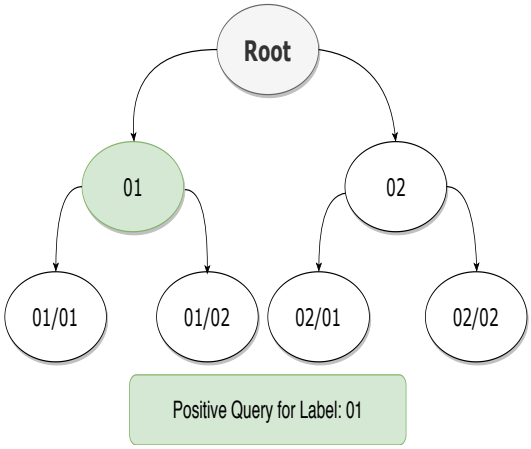
(b) Label 02/01 answered as negative

(c) Label 02/02 answered as negative

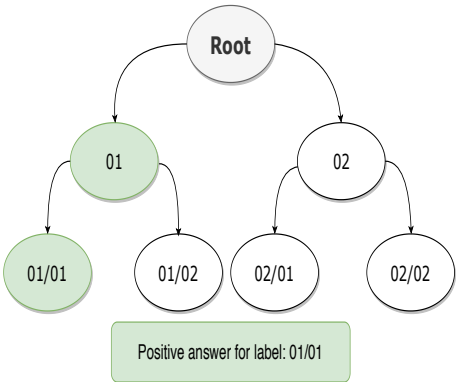
Fig. 5: Labelling procedure used by HALC (Yan and Huang, 2018)

At first, all ancestors are incorporated into the query in accordance with the hierarchy constraint, resulting in $[(U_q, 01),(U_q, 01/01),(U_q, 02),(U_q, 02/01),(U_q, 02/02)]$. Mind that the addition of labels is performed during the building of the query, thus the *Batchsize* parameter must be respected. Following the same answers from the previous example, the label 01 is labelled as positive (Figure 6a). Next, the label 01/01 is also labelled as positive (Figure 6b). Lastly, label 02 is negative, making its whole subtree (02/01 and 02/02) negative as well, and ending the procedure since all labels are answered already.

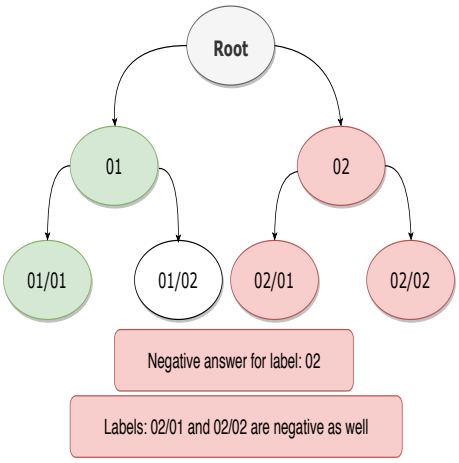
In this way, less budget is spent: as an ancestor label is negative, it is unnecessary to continue the labelling procedure up to the label requested. In the given example, the resulting cost is 7, which is considerably lower than with the method from Yan and Huang (2018). Additionally, in real applications, the oracle would be



(a) Label 01 answered as positive



(b) Label 01/01 answered as positive



(c) Label 02 answered as negative

Fig. 6: Proposed labelling procedure

more efficient, since it would often have to answer fewer (and easier, thus cheaper) labels. Following the hierarchy can also be considered more similar to what humans usually do. Moreover, some real world applications demand knowledge from ancestors labels to correctly answer the label requested. For instance, in an image annotation task, a query $(image, poodle)$ would be implicitly proceeded by a query $(image, dog)$.

Since the oracle provides answers only for specific labels, the augmented training sets are likely to contain partially labelled instances. Thus, it is mandatory to employ HMC predictive models capable of handling such instances, i.e. instances whose labels are not completely known. This restriction is promptly solved by employing methods from the local approach. For instance, Clus-HSC, a local method based on predictive clustering trees (Vens et al., 2008), trains a model for each edge of the hierarchy. Hence, whenever freshly labelled data is obtained, only those models that received new data points should be re-trained.

To summarize the previous sections, H-QBC builds queries containing the instance-label pairs from the unlabelled dataset with highest values for the Equation 2. Next, it includes the ancestors of the selected pairs in accordance with the hierarchy constraint and the maximum number of instance-label pairs ($BatchSize$) parameter. The building of the query is finished immediately if $BatchSize$ is reached. Finally, the oracle answers queries following the procedure described in Figure 6, adding extra label information because of the hierarchy constraint. As requirements, the HMC method used must be able to handle partially labelled instances, and it should also consist of an ensemble of classifiers since we are employing the variance of prediction probabilities.

3.4 Algorithm

Algorithm 1 describes in details how our method works. At first, a model is built using the initial labelled dataset **L**, and the available budget is set to 0. Similarly, the subset of new data points (NewData) is set to empty, and the amount of budget available (including possible leftovers from previous iterations) is increased by a fixed amount defined by B (budget per iteration).

Afterwards, the cycle begins and a number of data points ($BatchSize$) is selected using Equation 2 (Line 6). At Line 7, the labelling process by the Oracle starts. For every DataPoint, we verify if it is affordable according to the BudgetIteration available (Line 8) since labels located in different levels of Y might have different costs. In a positive case, the Oracle provides the answer for the data point (Line 9), and its cost is subtracted from the budget (Line 10).

If a label is answered as negative, the descendants of the label must be negative as well. Consequently, labels can be automatically answered by applying the hierarchy constraint (Line 11), and must be removed from the query (Line 12).

After obtaining all the labels, NewData is added to the labelled dataset (Line 15), and removed from the unlabelled dataset (Line 16). Hence, data points contained in NewData are not considered in the next iterations. Mind that an instance is removed from the unlabelled dataset only when it is completely answered. Finally, a new predictive model is built in Line 17 using the new labelled dataset, and subsequently evaluated in Line 18. If the stopping criterion is not met, the cycle is repeated.

4 Experimental methodology

In this section, we present details about the methodology used in this work. First, we provide information about the datasets employed. Next, we present which active

Algorithm 1: Hierarchical query-by-committee

Data: **L:** Labelled Data
U: Unlabelled Data
Y: Hierarchy of labels
B: Budget
c: Cost vector
BatchSize: Query size
Result: Model trained using selected data points

```
BudgetIteration ← 0 ;
Model ← f(L);
while not stopping condition do
    NewData ← ∅ ;
    BudgetIteration ← BudgetIteration + B ;
    Query ← H-QBC(Model,U,BatchSize);
    foreach DataPoint in Query do
        if cost(DataPoint,c) ≤ BudgetIteration then
            LabelledDataPoint ← Label(DataPoint,Oracle) ;
            BudgetIteration ← BudgetIteration - cost(DataPoints,c) ;
            NewDataPoints ← HierarchyConstraint(LabelledDataPoints,Y)
             ;
            Query ← Query \ NewDataPoint ;
            NewData ← NewData ∪ NewDataPoint ;
        end
        L ← L ∪ NewData ;
        U ← U \ NewData ;
        Model ← f(L); /* rebuild model                              */
        Evaluate(Model); /* update evaluation criterion           */
    end
end
return Model;
```

learning algorithms were implemented and the base classifiers used. Finally, we describe evaluation measures used in the experimental setup.

4.1 Datasets

In this work, we have used publicly available benchmark datasets from previous studies[3][4][5] on HMC. These datasets have been vastly used in the HMC literature (Vens et al., 2008; Schietgat et al., 2010; Cerri et al., 2016). According to Pliakos and Vens (2018), some of them suffer from non-unique feature representations, which may hinder and possibly misguide experiments. Hence, we have used only the datasets recommended by (Pliakos and Vens, 2018).

[3] https://dtai.cs.kuleuven.be/clus/hmcdatasets/
[4] https://dtai.cs.kuleuven.be/clus/hmc-ens/
[5] http://kt.ijs.si/DragiKocev/PhD/resources/doku.php?id=hmc_classification

The datasets used in Yan and Huang (2018) were adapted to facilitate active learning. More specifically, the authors have removed labels with few instances. In this work we have decided to maintain the datasets in their original form.

Originally, these datasets are presented in three subsets: training, validation and testing. In active learning scenarios, however, the initial amount of labeled data is considerably low, and consequently, a validation dataset may not be affordable. Thus, we have merged both train and validation subset and evaluated the performance using the test subset. Further, in a similar fashion to other AL work (Chakraborty et al., 2015; Wu et al., 2018), we randomly selected a small number of instances from the merged training dataset, corresponding to 10%, as the initial labelled dataset. For the other 90% we have removed their labels, making them the unlabeled dataset.

Table 3 presents statistical information about the datasets employed. All datasets used this work have a tree-shaped hierarchy. Following the order presented in the Table 3, the datasets from Cellcycle to Seq are related to protein function prediction of yeast species (*Saccharomyces cerevisiae*), whereas the Seq_ara, Scop_ara, Exprindiv_ara and Interpro_ara datasets are associated to plant species (*Arabidopsis thaliana*). Differently from that, Enron (Klimt and Yang, 2004) and Reuters (Lewis et al., 2004) are related to text categorization.

Dataset	#Labeled	#Unlabeled	#Test	#Features	#Hierarchical Nodes	#Levels
Cellcycle	248	2228	1281	77	499	6
Derisi	245	2205	1275	63	499	6
Expr	250	2244	1291	551	499	6
Eisen	159	1428	837	79	461	6
Gasch1	248	2232	1284	173	499	6
Gasch2	249	2239	1291	52	499	6
Seq	246	2209	1264	4450	263	4
Spo	244	2193	1266	80	499	6
Seq_ara	206	1849	1042	4450	250	4
Scop_ara	246	2209	1264	2003	263	4
Exprindiv_ara	232	2082	1182	1251	261	4
Interpro_ara	246	2209	1264	2815	263	4
Enron	99	889	660	1001	48	2
Reuters	300	2700	3000	55	101	2

Table 3: Datasets statistics information

There is a considerable difference in hierarchy size, since the smallest one, Enron, contains only 48 classes, whereas many protein function prediction datasets have up to 499 classes. That may seem relatively small, for instance when compared to extreme multi-label datasets (Gargiulo et al., 2019), nonetheless they are still of considerable size considering that they are tree-shaped. There is also a discrepancy between the number of features, as it ranges from 77 up to 4450. Additionally, the number of the levels of the hierarchy ranges from 2 to 6. Other taxonomies structured as directed acyclic graphs, such as the Gene Ontology, can present more possible classes, with the additional characteristic that one class can have more than one parent class in the hierarchy. Such taxonomies were investigated by works such as (Valentini, 2010; Yu et al., 2017; Nakano et al., 2019; Zhao et al., 2020), nonetheless we consider them out of scope for this work.

4.2 Evaluation Measures

In accordance with most of the work in hierarchical multi-label classification, we have adopted the Pooled Area Under the precision-recall curve (AUPRC) as the evaluation measure (Vens et al., 2008; Cerri et al., 2015, 2016). The Pooled AUPRC corresponds to the area under the precision-recall curve generated by taking the micro-average of precision and recall over all classes for different threshold values. In the equations below, tp stands for true positive, fp means false positive, fn refers to false negative and i ranges over all classes.

$$Precision = \frac{\sum tp_i}{\sum tp_i + \sum fp_i} \quad (3) \qquad Recall = \frac{\sum tp_i}{\sum tp_i + \sum fn_i} \quad (4)$$

Moreover, most of the works evaluate AL algorithms by measuring how fast the performance of the model increases throughout iterations, usually based on visual inspection. In an attempt to remove subjectivity, we decided to measure the area under the curve generated by plotting the PooledAUPRC values (y-axis) obtained using all the labelled data available at a certain iteration (x-axis). Intuitively, superior AL algorithms provide more informative data, consequently making the curve raise faster and providing higher values for the area under it.

The formulation is presented in Equation 5, where k represents an iteration, Query_k corresponds to the number of labels queried at iteration k, PooledAUPRC_k is the value PooledAUPRC obtained at iteration k and area is a function which provides the area under the trapezium defined by Query_k, $\text{PooledAUPRC}_{k-1}$and PooledAUPRC_k.

$$\sum_k \text{area}(\text{Query}_k, \text{PooledAUPRC}_{k-1}, \text{PooledAUPRC}_k) \quad (5)$$

Moreover, in order to provide statistical evidence, we have used the Friedman-Nemenyi tests as suggested by Demsar (2006). At first, the Friedman test ranks the evaluated algorithms where the best performing method is ranked first, the second best method is ranked second, etc. It also determines whether there is a statistically significant difference. In case of a positive outcome, the Nemenyi test determines which algorithms are statistically significant different from each other, by comparing the difference in their rankings to a critical distance. Graphically, methods connected by a horizontal bar are not statistically significantly different.

4.3 Base Classifiers: Clus-HSC

In this work we should employ a local approach since we are considering Label-Based methods. We have employed the Clus-HSC method with an ensemble of predictive clustering trees as local classifiers (Vens et al., 2008; Schietgat et al., 2010) for all AL algorithms methods evaluated.

The rationale behind this relies on three aspects:

- HMC benchmark datasets have many missing attribute values which can be handled by decision trees. Other base classifiers, such as SVMs, would require data imputation which is not trivial and has not been addressed in HMC;

- Using different classifiers for each AL method will result in incomparable values for their areas (Equation 5), since their performance (Pooled AUPRC) would differ from each other;
- Our method demands a committee of local classifiers;

As the ensemble method, we have employed the Random Forest setting due to its robustness against overfitting, and overall satisfactory results. As for the parameters, we have followed some of the recommendations by Schietgat et al. (2010); Kocev et al. (2013). For all experiments, we have employed 50 trees in the forest, a minimum of 5 instances per leaf node, $log_2(|attr|) + 1$ random attributes available for each split where $|attr|$ corresponds to the total number of attributes and an F-test stopping criterion with significance level 0.05.

4.4 Active learning Algorithms

In the following, we report the baseline label-based algorithms that we have implemented within Clus. We compare our method against baseline methods: random, uncertainty sampling and query-by-committee. Additionally, we also compare our method against recent methods from the literature: SSMAL and HALC.

- Random (selects instance-label pairs without any criteria, average of 5 executions);
- Uncertainty sampling: baseline algorithm which selects data points located close to the decision border (Equation 6) (Lewis and Catlett, 1994);
- Query-by-committee: baseline algorithm which selects data points using Equation 1) (Seung et al., 1992). This comparison is included, since our proposed algorithm is an extended version of it;
- Semi-supervised multi-label active learning (Ye et al., 2015b; Wu et al., 2017);
- Hierarchical active learning with cost (Yan and Huang, 2018).
- Hierarchical query-by-committee: our proposed method (Equation 2);

All algorithms were evaluated using the labelled procedure described in Section 3 (Figure 6), with exception of HALC which has its own procedure (Figure 5). Likewise, all AL algorithms are using random forests as base-classifiers.

The semi-supervised multi-label active learning (SSMAL) is a label-based method recently proposed for non-hierarchical multi-label classification which combines label correlation and uncertainty sampling (Equation 6). Equation 7 presents how the label correlation is calculated. In this case, y_i and y_j correspond to labels from the hierarchy, A corresponds to the numbers of instances where both labels appear, D corresponds to the number of instances where both labels do not occur, B corresponds to the number of instances where y_i appears, but y_j does not. Lastly, C corresponds to the number of instances where y_j appear, but y_i does not. We present its formulation below:

$$Unc(\mathbf{x}_i, y_j) = |P(y_j|\mathbf{x}_i) - 0.5| \tag{6}$$

$$M(y_i, y_j) = \frac{AD - BC}{\sqrt{(A+B) * (A+C) * (C+D) * (B+D)}} \tag{7}$$

Following, Equation 7 is extended in Equation 8 to consider the correlations among instances from the unlabeled dataset, assuming that some of them may not be entirely labelled. In this case, l_u corresponds to the number of unknown labels of a particular instance.

$$MU(x_i, y_j) = \begin{cases} \frac{1}{l_u} \sum_{k=1}^{l} |M_{jk}| \times sign(y_k \in U) \text{ if } l_u \geq 1 \\ 0 \text{ if } l_u = 1 \end{cases} \tag{8}$$

Finally, Equation 8 is combined with Equation 6 to form the final selection criterion. Mind that SSMAL prioritizes Instance-Label pairs with lower values.

$$SSMAL(\mathbf{x}_i, y_j) = \gamma * Unc(\mathbf{x}_i, y_j) - (1 - \gamma) MU(\mathbf{x}_i, y_j) \tag{9}$$

Its weighting parameter (γ), responsible to balance the uncertainty and the label correlation, was set to *0.9* as the authors pointed out to be the best. Moreover, since we are not considering the label inferring procedure, SSMAL is equivalent to CSMAL (Ye et al., 2015b).

Likewise, hierarchical active learning with cost (HALC) is the current state-of-the-art method in active learning for hierarchical multi-label classification. In its formulation, HALC uses the evolutionary optimization algorithm POSS which requires a number of iterations (*IterationsNumber*), and a population size (*PopulationSize*), as parameters. We have fixed *IterationsNumber* to 100 and *PopulationSize* to 50. Note that these parameter values were not reported in Yan and Huang (2018), and the datasets addressed by the authors were significantly smaller. Differently from that, our method, H-QBC, does not require the tuning of any parameter.

AL algorithms for hierarchical single label classification are not included because the current state-of-the-art, BatchRank (Chakraborty et al., 2015), is an instance-based method.

4.4.1 Batch size and budget

Since we are interested in seeing how the algorithms converge throughout the experiments, we did not limit the number of iterations. Hence, the algorithms will select up to a certain number (defined as the batch size) of instance-label pairs at every iteration until the unlabelled dataset is completely labelled.

Unfortunately, the literature does not have a standard value for the batch size. If a small batch size is used, AL algorithms are likely to select more informative data-points, nonetheless it might be inefficient in real applications since the model will take longer to improve, and the oracle be might required to answer numerous queries.

Some studies select a fixed number of pairs based on the size of dataset (Wu et al., 2017, 2018), e.g. $5 \times l$ or $10 \times l$ where l is the number of possible labels in each dataset. Thus, optimizing the batch size is an unexplored area, and might be investigated in future work.

Using a similar value, we have set batch size to 2500 per iteration which roughly corresponds to $10 \times l$ for the *Arabidopsis thaliana* datasets, and $5 \times l$ for the *Saccharomyces cerevisiae* datasets.

As for the budget, most of the studies do not evaluate it, meaning that an infinite budget is available at every iteration. Differently from that, in this work, we have limited the budget to 2500 per iteration.

The cost of each label may vary according to the application. For instance some labels can become more expensive according to their depth in the hierarchy, e.g. the deeper the more expensive. Thus, we have used the different cost vectors: [1,1,1,1,1,1], [1,3,6,9,12,15] and [1,5,10,15,20,25,30], assuming hierarchies with 6 levels. Intuitively, each value in the vector corresponds to the cost of a label on a specific level of the hierarchy. The unitary cost vector is included since there might exists cases where labels have the same cost despite their location in the hierarchy, whereas the two other cost vectors were imported from (Yan and Huang, 2018). In smaller hierarchies, the cost vector is clipped to the maximum depth.

5 Results and Discussion

In this section, we present our experiments. At first, we provide a predictive performance comparison between the algorithms, followed by an evaluation on how our algorithm differs from its baseline counterpart. Next, we compare the time-efficiency of all evaluated algorithms. Lastly, we discuss the contents of each query, showing the mean depth of the labels and their relation to the budget. All graphs presented here are best seen in colours.

5.1 Comparison between H-QBC and algorithms from the literature

Even tough we have performed experiments using 3 cost vectors, we present the curves using the unitary cost vector [1,1,1,1,1,1]. The other curves can be found at the Supplementary Material (Section 1.1). Moreover, we have randomized the initial labelled dataset three times for the experiments involving the unitary cost vector and presented the average values.

In Figures 7, we show the curves, obtained by plotting the performance (Pooled AUPRC) on the y-axis and the number of queried answered on the x-axis, throughout iterations using different cost vectors. Moreover, Table 4 contains the mean area under the curves obtained via Equation 5. Finally, in Figures 8, 9 and 10, we show the Friedman-Nemenyi tests obtained using the area under the curve and the three different cost vectors.

As can be seen in the curves, our algorithm H-QBC managed to outperform its competitors in most of the cases. Its curve (orange) raises faster than the other algorithms, showing that the instance-label pairs requested are indeed more informative. In many datasets, the difference is visually noticeable since the other methods require more pairs to match H-QBC's performance.

Additionally, the area under the curves (Table 4), demonstrate thats H-QBC provided higher values. This is further reinforced by the Friedman-Nemenyi tests shown in Figures 8, 9 and 10, where H-QBC was constantly ranked in the highest position. Hence, our method is capable of providing superior results in datasets with different characteristics and different label costs, nonetheless it is not statistically significantly different from the other methods.

Only in the Enron (Figure 7n) dataset using the cost vector [1,1,1,1,1,1], HALC provided a slightly faster increase in the performance since its curve (black) raises before H-QBC's (orange), nonetheless this advantage was very minimal, and HALC is still inferior when considering all iterations, as shown in Table 4. The Enron dataset, as shown in Table 3, is fairly smaller when compared to the others, similarly to the datasets used in the original work (Yan and Huang, 2018). However, it is worth mentioning that the datasets in HMC are usually much larger than Enron, nonetheless HALC is superior when compared to the baselines query-by-committee and uncertainty sampling. HALC was ranked in the third position for the unitary cost vector and second when using the cost vector [1,3,6,9,12,15], nonetheless it moved to the fourth position when using [1,5,10,15,20,25] being no longer different than Random.

Similarly, the other competitor method, SSMAL, also provided better results than the baselines. However, it still managed to be ranked higher than HCAL when using the cost vectors [1,1,1,1,1,1] and [1,5,10,15,20,25], as shown in Friedman-Nemenyi test in Figures 8 and 10. These results are interesting since SSMAL considers the relationship between any pair of labels that co-occur in instances (Equation 8), whereas H-QBC considers only ascendants and descendants. We suspect that this difference in the measure for label correlation is responsible for the difference in performance as well.

The Random algorithm, as expected, underperformed in most of the cases. Random provided a steep increase in performance in the beginning, but then failed to query valuable data points as the performance increased linearly in almost all datasets.

Apart from that, it is noticeable that in most of the datasets the performance approaches the Bound very rapidly, with only few instance-label pairs. We consider this finding motivating, as it proves that there exists an optimal way of selecting data points, attesting the applicability of AL in this context.

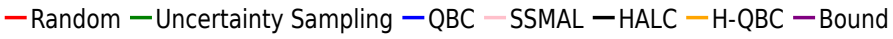
On the other hand, after its quick raise, the performance starts increasing very slowly, and sometimes even decreasing, as seen in Scop_ara dataset. This is not necessarily an indicator that the algorithms are failing, but rather running out of meaningful instance-label pairs. In a real application, such oscillation might indicate a stopping criterion where labelling more data points is not profitable.

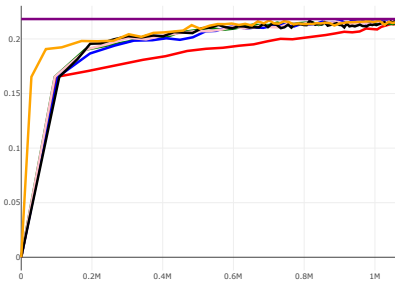
5.2 Time-wise comparison

In a complementary way, we also measured the time efficiency of each algorithm. In Table 5, we present the average time, measured in seconds, to build queries for each algorithm.

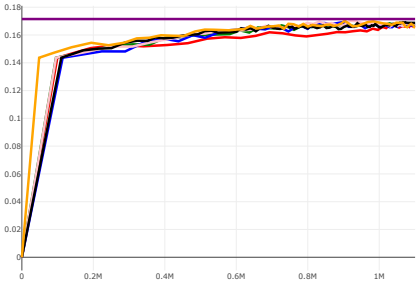
Since the budget does not affect the query building time, we are showing the time using the cost vector [1,5,10,15,20,25]. As can be seen in Table 5, our algorithm, H-QBC, is noticeably faster than the current state-of-art, HALC. In many cases, HALC needs more than 1000 seconds to build its query, whereas H-QBC's is approximately 10 times faster where the mean computational time is around 100 seconds. In real applications, such difference may be relevant, considering the volume and velocity of the data.

Apart from that, the baseline algorithms are still faster than our method. Such result is expected, since the baselines were designed to traditional classi-

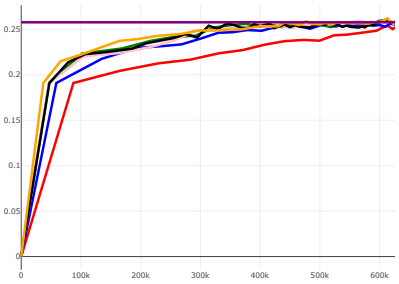




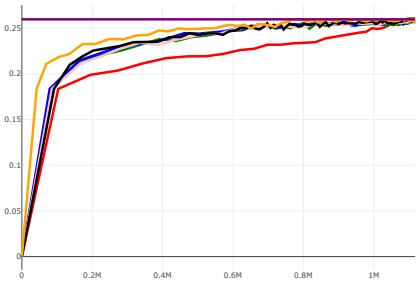
(a) Cellcycle



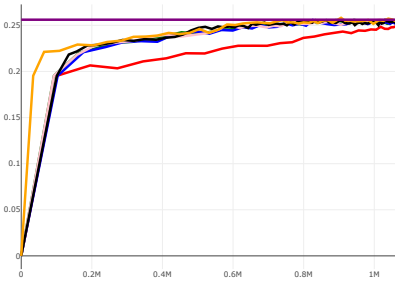
(b) Derisi



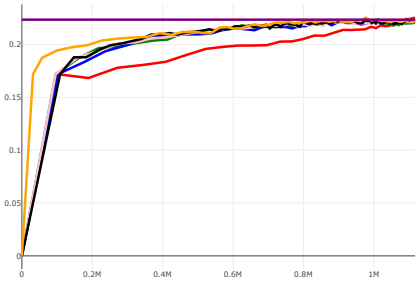
(c) Eisen



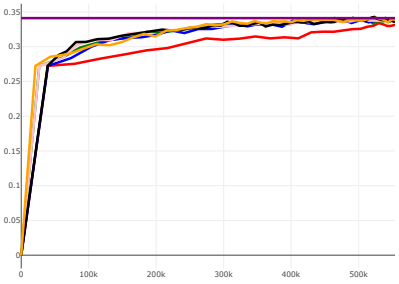
(d) Expr



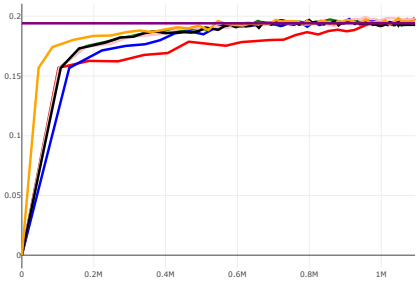
(e) Gasch1



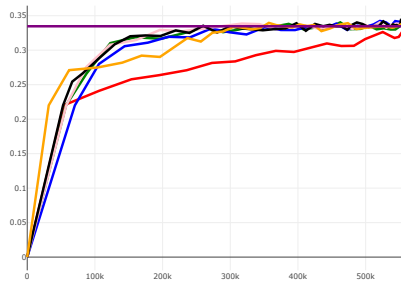
(f) Gasch2



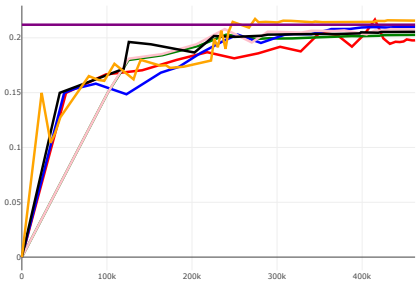
(g) Seq



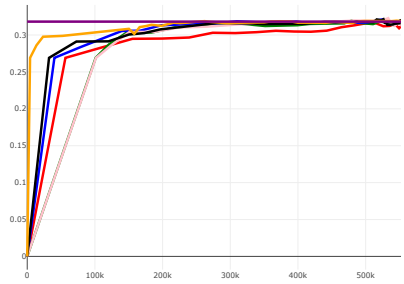
(h) Spo



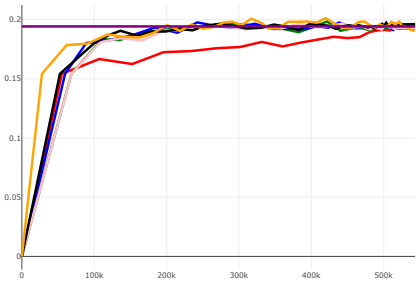
(i) Seq_ara



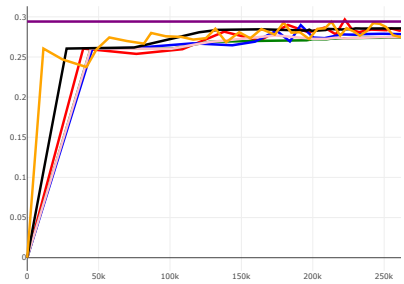
(j) Scop_ara



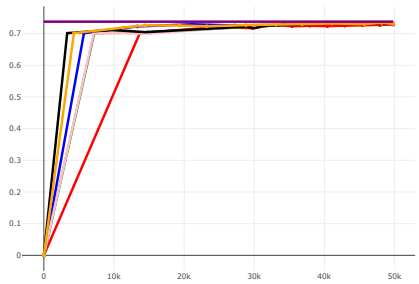
(k) Interpro_ara



(l) Exprindiv_ara



(m) Reuters



(n) Enron

Fig. 7: Curves obtained using the cost vector [1,1,1,1,1,1]. The x-axes correspond to the number of labels queried, whereas the y-axes correspond to the PooledAUPRC for each iteration. Bound corresponds to an upper bound, i.e., the performance obtained with all the available data (unlabelled and labelled) used as the training set.

	Random	US	QBC	SSMAL	HALC	H-QBC
Cellcycle	214.5	227.21	226.62	227.66	227.17	**230.74**
Derisi	172.9	175.82	174.91	176.04	176.12	**177.25**
Eisen	148.73	158.89	156.82	158.58	158.41	**160.5**
Expr	252.97	268.05	269.73	267.93	268.6	**276.48**
Gasch1	251.92	267.2	265.32	266.71	267.15	**270.86**
Gasch2	221.95	234.03	233.52	234.07	234.09	**237.61**
Seq	180.52	189.53	187.97	**189.63**	189.4	189.31
Spo	192.69	203.22	201.22	202.9	203.36	**205.39**
Seq_ara	166.56	181.85	180.36	**182.9**	182.71	180.46
Scop_ara	86.72	88.25	88.0	88.72	**89.68**	86.73
Exprindiv_ara	95.9	102.14	102.37	102.23	102.22	**102.93**
Interpro_ara	174.2	178.38	180.98	178.24	179.83	**181.99**
Reuters	81.78	80.77	81.8	81.99	**82.14**	81.99
Enron	35.79	35.99	36.14	36.01	35.89	**36.11**

Table 4: Mean Area under the curve (generated throughout the iterations) using the cost vector [1,1,1,1,1,1] and 3 randomized initial labelled datasets. All values were multiplied by a 10^{-3}

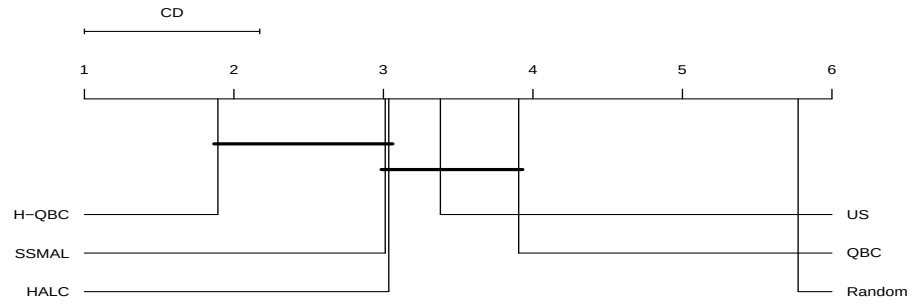


Fig. 8: Friedmann-Nemenyi test evaluating the area under the curve throughout iteration using the vector cost [1,1,1,1,1,1].

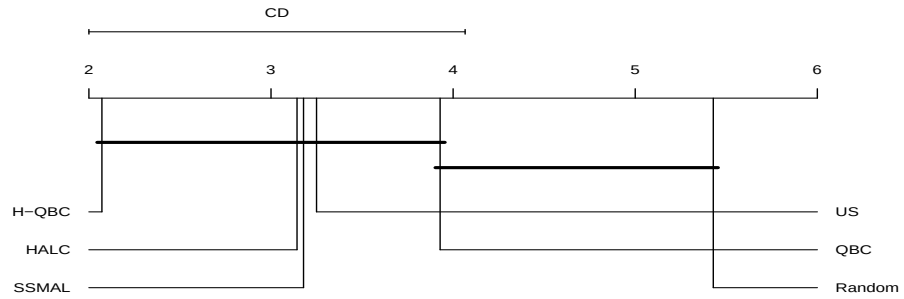


Fig. 9: Friedmann-Nemenyi test evaluating the area under the curve throughout iteration using the vector cost [1,3,6,9,12,15].

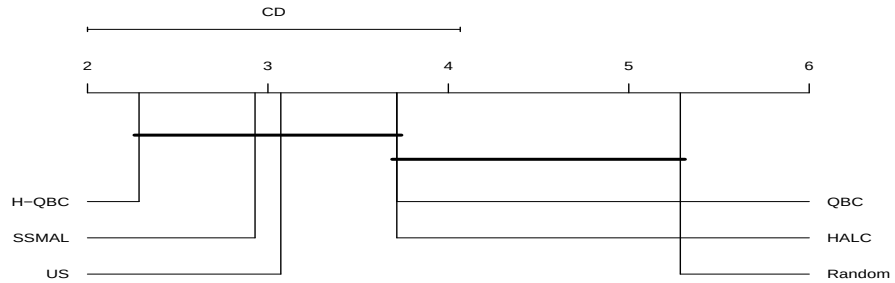


Fig. 10: Friedmann-Nemenyi test evaluating the area under the curve throughout iteration using the vector cost [1,5,10,15,20,25].

fication problems where there is no need to consider the underlying hierarchy. Consequently, although they are more efficient, their performance is inferior.

Moreover, we could notice that the time scales considerably according to the hierarchy size. In datasets with smaller hierarchies for all methods, such as Enron and Reuters, the query building time is considerably shorter. However, HALC's time complexity scales considerably in datasets with higher number of classes (hierarchy size), as seen in the experiments with the Cellcycle dataset, for instance.

	US	QBC	SSMAL	HALC	H-QBC
Cellcycle	25.74	30.49	91.21	1287.3	139.68
Derisi	25.57	40.08	77.58	1598.68	73.36
Eisen	21.01	19.9	58.73	737.71	93.1
Expr	28.65	27.74	100.01	1790.72	74.5
Gasch1	38.83	24.97	98.42	1499.69	66.94
Gasch2	25.85	25.52	89.72	1505.29	61.78
Seq	26.48	25.66	38.92	798.06	126.28
Spo	24.59	24.51	86.8	1547.68	74.97
Seq_ara	21.35	21.05	28.93	689.47	52.48
Scop_ara	12.53	14.05	18.79	547.29	85.77
Exprindiv_ara	18.28	19.12	26.82	787.28	91.26
Interpro_ara	14.75	16.69	24.69	756.18	86.99
Reuters	13.64	14.37	14.64	209.48	69.26
Enron	8.6	9.84	9.5	39.39	45.66

Table 5: Average time to build one query measured in seconds considering the vector cost [1,5,10,15,20,25].

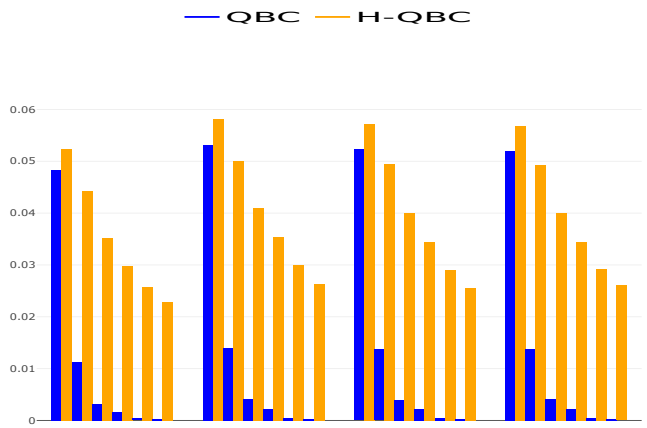


Fig. 11: The y-axis contains the mean variance values obtained using QBC (Equation 1) and H-QBC (Equation 2) for labels located in each of 6 hierarchy levels of the Cellcycle dataset. Each bar represents a level of the hierarchy, and bars are grouped by iteration (x-axis).

5.3 Comparison of variance values through the hierarchy levels

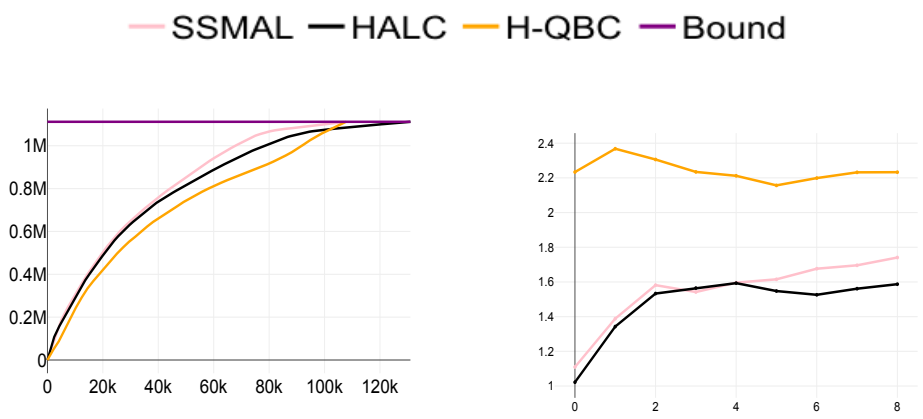
In order to further evaluate our algorithm, we present a comparison between H-QBC and its baseline QBC. More specifically, we investigated their variance values and how they change according to the hierarchy level.

The variance (Equation 1) is expected to be low in deeper levels of the hierarchy. As can be seen in Figure 11, this was confirmed since QBC obtained inferior variance values overall. In fact, QBC yielded relatively high values for the first level (around 0.05), nonetheless it drops substantially in the next levels, reaching almost 0 in the deepest level. Such value demotivates the selection of deep instance-label pairs, leading to a Query with plenty of labels from the first level, and possibly worse results. On the other hand, H-QBC consistently provided higher values, specially for deeper labels.

We can also notice that the variance in the all levels increased for both algorithms after the first iteration. This finding is rather unexpected since the first level is technically the easiest to predict, and the model should provide confident predictions despite the lack of data. The variance from the other levels did not visibly change throughout the iterations.

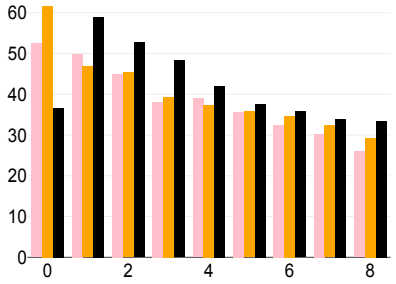
5.4 Comparison between the selected instance-label pairs

Lastly, we further analyse the instance-label pairs selected by the three AL algorithms that performed best in our experiments. More specifically, we focus on the Cellcycle dataset, using the unitary cost vector ([1,1,1,1,1,1]), and the H-QBC, SSMAL and HALC algorithms. Graphs associated to the Seq_ara and Reuters can be found in the Supplementary Material (Section 1.2).



(a) Comparison between the the number of labels obtained throughout all iterations (y-axis) and the budget spent (x-axis).

(b) Mean depth (y-axis) of the label-pair selected in the first iterations (x-axis).



(c) Percentage of positive answers (y-axis) obtained in the first iterations (x-axis)

Fig. 12: Analysing the instance-label pairs selected by H-QBC, SSMAL and HALC using the cost vector [1,1,1,1,1,1] in the Cellcycle dataset.

As can be seen in Figure 12a, there is a considerable difference between the number of labels obtained (including the extra labels answered via the hierarchy constraint) and the budget spent between the algorithms. In most of the cases, SSMAL, followed by HALC and H-QBC, obtains more labels while spending less budget since negative queries result in an entire negative subtree due to the hierarchy constraint. This finding is rather surprising as H-QBC provides better results in general, and thus, it may be expected to provide more labels for less budget. However, obtaining more labels is not necessarily associated to better results, since these instance-label pairs might not be informative.

As shown in Figure 12b, H-QBC prioritizes instance-label pairs that belong to deeper levels in the hierarchy, whereas SSMAL and HALC focus on pairs from the first levels. More specifically, H-QBC includes pairs from the second level on, and, differently from that, SSMAL and HALC prioritize labels from the first level.

We believe that selecting pairs only from the first level may not be appropriate since such AL algorithms are ignoring most of the labels in the hierarchy. This difference also results in SSMAL and HALC obtaining more labels per budget spent, as seen in Figure 12a, since a negative answer in the first level results in a entire negative subtree.

Moreover, there is a difference between the percentage of positive answers provided by the oracle, as seen in Figure 12c. H-QBC obtains more positive answers in the first iteration, and HALC obtained more positives in the next iterations. Hence, despite prioritizing deeper labels, H-QBC still obtains a reasonable number of positive answers. This is particularly interesting in the HMC context because most of the datasets are likely to have instances with very few positive labels, making them rare and more difficult to identify with AL algorithms.

To summarize this, our algorithm, H-QBC, provides better predictive performance in general, while prioritizing instance-label pairs in deeper levels, resulting in more informative queries, but fewer labels answered in total considering the hierarchy constraint.

6 Conclusion

In this work, we have studied active learning applied to hierarchical multi-label classification (HMC). HMC brings new challenges to active learning, as the datasets usually have high number of labels, alongside an underlying hierarchy which defines relationships among the classes. In order to perform reproducible research, we have developed an active learning framework which incorporates baseline algorithms and state-of-the-art active learning methods from related classification tasks [6].

Moreover, we also presented a new AL algorithm suitable for HMC, validated on 14 datasets from different domains. When compared to the current state-of-art, our algorithm, is capable of providing superior results, while being more computationally efficient. Additionally, it can also improve the performance faster, in the expense of less budget.

Overall, our results attested that HMC problems can be handled in an AL setting since most of the algorithms are superior to the random approach, meaning that there are suitable strategies to select data points in this context.

In future work, we intend to incorporate datasets whose hierarchies are structured as direct acyclic graphs, such as the Gene Ontology in protein function prediction, using the new datasets investigated by Nakano et al. (2019). In these cases, classes are allowed to have more than one superclass, necessitating AL algorithms capable of considering many hierarchical paths from the root node to any class. Additionally, the literature still lacks studies on AL applied to multi-target regression (Borchani et al., 2015) and hierarchical regression (Mileski et al., 2017), which are addressable by extending the framework presented.

Moreover, we have only considered variance as the disagreement measure due to its simplicity. Future work might investigate new measures, specially measures that do not involve an ensemble method. On top of that, an improved version could

[6] https://itec.kuleuven-kulak.be/supportingmaterial

also include a criterion regarding relationships between classes that are located far in the hierarchy, i.e. not in the same sub-tree of the hierarchy.

Additionally, recent active learning studies in non-hierarchical multi-label classification are performing label inference (Wu et al., 2017, 2018), an extra step in the AL framework which allows the automatic labelling of certain labels, reducing even further the overall cost. After consulting the oracle, the updated model is used to infer some instance-label pairs. This idea is identical to self-training from the semi-supervised learning field. We would like to incorporate this idea to the HMC context as well.

Finally, recently a semi-supervised version of predictive clustering trees was proposed by Levatić et al. (2018), nonetheless it was only validated on multi-target regression datasets. Future work might compare its performances against active learning algorithms since it can handled partially labelled instances.

Acknowledgements

We are grateful to FAPESP (grants #2016/12489-2 and #2017/19264-9), Research Fund Flanders (FWO) and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 for providing financial support.

References

Athanasopoulos, G., Gamakumara, P., Panagiotelis, A., Hyndman, R. J., and Affan, M. (2020). *Hierarchical Forecasting*, paginas 689–719. Springer International Publishing, Cham.

Bekker, J. and Davis, J. (2018). Learning from positive and unlabeled data: A survey. *CoRR*, abs/1811.04820.

Borchani, H., Varando, G., Bielza, C., and Larrañaga, P. (2015). A survey on multi-output regression. *Wiley Int. Rev. Data Min. and Knowl. Disc.*, 5(5):216–233.

Brinker, K. (2006). On active learning in multi-label classification. In *From Data and Information Analysis to Knowledge Engineering*, paginas 206–213. Springer.

Cerri, R., Barros, R., and de Carvalho, A. (2015). Hierarchical classification of gene ontology-based protein functions with neural networks. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, paginas 1–8.

Cerri, R., Barros, R. C., and de Carvalho, A. C. P. L. F. (2012). A genetic algorithm for hierarchical multi-label classification. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, paginas 250–255, New York, NY, USA. ACM.

Cerri, R., Barros, R. C., P. L. F. de Carvalho, A. C., and Jin, Y. (2016). Reduction strategies for hierarchical multi-label classification in protein function prediction. *BMC Bioinformatics*, 17(1):373.

Cerri, R., Basgalupp, M. P., Barros, R. C., and de Carvalho, A. C. (2019). Inducing hierarchical multi-label classification rules with genetic algorithms. *Applied Soft Computing*, 77:584 – 604.

Chakraborty, S., Balasubramanian, V., and Panchanathan, S. (2011). Optimal batch selection for active learning in multi-label classification. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, paginas 1413–1416, New York, NY, USA. ACM.

Chakraborty, S., Balasubramanian, V., Sankar, A. R., Panchanathan, S., and Ye, J. (2015). Batchrank: A novel batch mode active learning framework for hierarchical classification. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, paginas 99–108. ACM.

Cheng, Y., Chen, Z., Fei, H., Wang, F., and Choudhary, A. (2014). Batch mode active learning with hierarchical-structured embedded variance. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, paginas 10–18. SIAM.

Cheng, Y., Zhang, K., Xie, Y., Agrawal, A., and Choudhary, A. (2012). On active learning in hierarchical classification. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, paginas 2467–2470. ACM.

Cherman, E. A., Papanikolaou, Y., Tsoumakas, G., and Monard, M. C. (2017). Multi-label active learning: key issues and a novel query strategy. *Evolving Systems*.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30.

Duin, J. D. (2017). *Hierarchical Active Learning Application to Mitochondrial Disease*. Tese de Doutorado, University of Nebraska.

Gargiulo, F., Silvestri, S., Ciampi, M., and Pietro, G. D. (2019). Deep neural network for hierarchical extreme multi-label text classification. *Applied Soft Computing*, 79:125 – 138.

Guo, A., Wu, J., Sheng, V. S., Zhao, P., and Cui, Z. (2017). Multi-label active learning with low-rank mapping for image classification. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, paginas 259–264.

Hoi, S. C. H., Jin, R., Zhu, J., and Lyu, M. R. (2006). Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pagina 417–424, New York, NY, USA. Association for Computing Machinery.

Huang, S., Jin, R., and Zhou, Z. (2014). Active learning by querying informative and representative examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(10):1936–1949.

Huang, S. and Zhou, Z. (2013). Active query driven by uncertainty and diversity for incremental multi-label learning. In *2013 IEEE 13th International Conference on Data Mining*, paginas 1079–1084.

Hung, C.-W. and Lin, H.-T. (2011). Multi-label active learning with auxiliary learner. In Hsu, C.-N. and Lee, W. S., editors, *Proceedings of the Asian Conference on Machine Learning*, volume 20 of *Proceedings of Machine Learning Research*, paginas 315–332, South Garden Hotels and Resorts, Taoyuan, Taiwain. PMLR.

Jiao, Y., Zhao, P., Wu, J., Xian, X., Xu, H., and Cui, Z. (2014). Active multi-label learning with optimal label subset selection. In Luo, X., Yu, J. X., and Li, Z., editors, *Advanced Data Mining and Applications*, paginas 523–534, Cham. Springer International Publishing.

Klimt, B. and Yang, Y. (2004). The enron corpus: A new dataset for email classification research. In *ECML '04: Proceedings of the 18th European Conference on Machine Learning – LNCS 3201*, paginas 217–226. Springer Berlin / Heidelberg.

Kocev, D., Vens, C., Struyf, J., and Džeroski, S. (2013). Tree ensembles for predicting structured outputs. *Pattern Recognition*, 46(3):817 – 833.

Krawczyk, B., Minku, L. L., Gama, J., Stefanowski, J., and Woźniak, M. (2017). Ensemble learning for data stream analysis: A survey. *Information Fusion*, 37:132–156.

Levatić, J., Ceci, M., Kocev, D., and Džeroski, S. (2017). Self-training for multi-target regression with tree ensembles. *Knowledge-based systems*, 123:41–60.

Levatić, J., Kocev, D., Ceci, M., and Džeroski, S. (2018). Semi-supervised trees for multi-target regression. *Information Sciences*, 450:109 – 127.

Lewis, D. D. and Catlett, J. (1994). Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the eleventh international conference on machine learning*, paginas 148–156.

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

Li, X. and Guo, Y. (2013). Active learning with multi-label svm classification. In *IJCAI International Joint Conference on Artificial Intelligence*, paginas 1479–1485.

Li, X., Kuang, D., and Ling, C. X. (2012). Active learning for hierarchical text classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, paginas 14–25. Springer.

Li, X., Ling, C. X., and Wang, H. (2013). Effective top-down active learning for hierarchical text classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, paginas 233–244. Springer.

Li, X., Wang, L., and Sung, E. (2004). Multilabel svm active learning for image classification. In *2004 International Conference on Image Processing, 2004. ICIP '04.*, volume 4, paginas 2207–2210 Vol. 4.

Mileski, V., Džeroski, S., and Kocev, D. (2017). Predictive clustering trees for hierarchical multi-target regression. In Adams, N., Tucker, A., and Weston, D., editors, *Advances in Intelligent Data Analysis XVI*, paginas 223–234, Cham. Springer International Publishing.

Mo, Y., Scott, S. D., and Downey, D. (2016). Learning hierarchically decomposable concepts with active over-labeling. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, paginas 340–349.

Nakano, F. K., Lietaert, M., and Vens, C. (2019). Machine learning for discovering missing or wrong protein function annotations. *BMC Bioinformatics*, 20(1):485.

Nakano, F. K., Pinto, W. J., Pappa, G. L., and Cerri, R. (2017). Top-down strategies for hierarchical classification of transposable elements with neural networks. In *International Joint Conference on Neural Networks (IJCNN)*, paginas 2539–2546.

Pliakos, K. and Vens, C. (2018). Mining features for biomedical data using clustering tree ensembles. *Journal of Biomedical Informatics*, 85:40 – 48.

Qi, G.-J., Hua, X.-S., Rui, Y., Tang, J., and Zhang, H.-J. (2008). Two-dimensional active learning for image classification. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, paginas 1–8.

Qian, C., Yu, Y., and Zhou, Z.-H. (2015). Subset selection by pareto optimization. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, paginas 1774–1782, Cambridge, MA, USA. MIT Press.

Reyes, O., Morell, C., and Ventura, S. (2018). Effective active learning strategy for multi-label learning. *Neurocomputing*, 273:494 – 508.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, paginas 1135–1144.

Rubens, N., Kaplan, D., and Sugiyama, M. (2011). Active learning in recommender systems. In Kantor, P., Ricci, F., Rokach, L., and Shapira, B., editors, *Recommender Systems Handbook*, paginas 735–767. Springer.

Schietgat, L., Vens, C., Struyf, J., Blockeel, H., Kocev, D., and Džeroski, S. (2010). Predicting gene function using hierarchical multi-label decision tree ensembles. *BMC Bioinformatics*, 11(1).

Settles, B. (2010). Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11.

Seung, H. S., Opper, M., and Sompolinsky, H. (1992). Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, paginas 287–294. ACM.

Štrumbelj, E. and Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665.

Valentini, G. (2010). True path rule hierarchical ensembles for genome-wide gene function prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(3):832–847.

van Engelen, J. E. and Hoos, H. H. (2019). A survey on semi-supervised learning. *Machine Learning*.

Vasisht, D., Damianou, A., Varma, M., and Kapoor, A. (2014). Active learning for sparse bayesian multilabel classification. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, paginas 472–481, New York, NY, USA. ACM.

Vens, C., Struyf, J., Schietgat, L., Džeroski, S., and Blockeel, H. (2008). Decision trees for hierarchical multi-label classification. *Machine Learning*, 73:185–214.

Wang, X., Zhao, H., and Lu, B.-l. (2011). Enhanced k-nearest neighbour algorithm for large-scale hierarchical multi-label classification. In *Proc Joint ECML/PKDD PASCAL Workshop on Large-Scale Hierarchical Classification*.

Wehrmann, J., Cerri, R., and Barros, R. (2018). Hierarchical multi-label classification networks. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, paginas 5075–5084, Stockholmsmässan, Stockholm Sweden. PMLR.

Wu, J., Guo, A., Sheng, V. S., Zhao, P., and Cui, Z. (2018). An active learning approach for multi-label image classification with sample noise. *International Journal of Pattern Recognition and Artificial Intelligence*, 32(03):1850005.

Wu, J., Sheng, V. S., Zhang, J., Zhao, P., and Cui, Z. (2014). Multi-label active learning for image classification. In *2014 IEEE International Conference on Image Processing (ICIP)*, paginas 5227–5231.

Wu, J., Ye, C., Sheng, V. S., Zhang, J., Zhao, P., and Cui, Z. (2017). Active learning with label correlation exploration for multi-label image classification. *IET Computer Vision*, 11(7):577–584.

Yan, Y. and Huang, S.-J. (2018). Cost-effective active learning for hierarchical multi-label classification. In *IJCAI*, paginas 2962–2968.

Yan, Y., Rosales, R., Fung, G., and Dy, J. G. (2011). Active learning from crowds. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pagina 1161–1168, Madison, WI, USA. Omnipress.

Yang, B., Sun, J.-T., Wang, T., and Chen, Z. (2009). Effective multi-label active learning for text classification. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, paginas 917–926, New York, NY, USA. ACM.

Yang, K., Ren, J., Zhu, Y., and Zhang, W. (2018). Active learning for wireless iot intrusion detection. *IEEE Wireless Communications*, 25(6):19–25.

Ye, C., Wu, J., Sheng, V., Zhao, P., and Cui, Z. (2015a). Multi-label active learning with label correlation for image classification. paginas 3437–3441.

Ye, C., Wu, J., Sheng, V. S., Zhao, S., Zhao, P., and Cui, Z. (2015b). Multi-label active learning with chi-square statistics for image classification. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, ICMR '15, pagina 583–586, New

York, NY, USA. Association for Computing Machinery.

Yu, G., Fu, G., Wang, J., and Zhao, Y. (2017). Newgoa: Predicting new go annotations of proteins by bi-random walks on a hybrid graph. *IEEE/ACM transactions on computational biology and bioinformatics*, 15(4):1390–1402.

Zeng, C., Zhou, W., Li, T., Shwartz, L., and Grabarnik, G. Y. (2017). Knowledge guided hierarchical multi-label classification over ticket data. *IEEE Transactions on Network and Service Management*, 14(2):246–260.

Zhang, B., Wang, Y., and Chen, F. (2014). Multilabel image classification via high-order label correlation driven active learning. *IEEE Transactions on Image Processing*, 23(3):1430–1441.

Zhang, B., Wang, Y., and Wang, W. (2012). Batch mode active learning for multi-label image classification with informative label correlation mining. In *2012 IEEE Workshop on the Applications of Computer Vision (WACV)*, paginas 401–407.

Zhang, M.-L. (2009). Ml-rbf: Rbf neural networks for multi-label learning. *Neural Processing Letters*, 29:61–74.

Zhang, Z., Zhang, J., Liu, Y., Wang, Z., and Deng, L. (2017). Ontological function annotation of long non-coding RNAs through hierarchical multi-label classification. *Bioinformatics*, 34(10):1750–1757.

Zhao, Y., Wang, J., Chen, J., Zhang, X., Guo, M., and Yu, G. (2020). A literature review of gene function prediction by modeling gene ontology. *Frontiers in Genetics*, 11:400.