Label Cluster Chains for Multi-Label Classification

Elaine Cecília Gatto[1*], Felipe Kenji Nakano[2], Jesse Read[3], Mauri Ferrandin[4], Ricardo Cerri[5], Celine Vens[2]

[1*]Department of Computer Science, Federal University of São Carlos, São Carlos, 13565-905, SP, Brazil.
[2]Faculty of Medicine, Katholieke Universiteit Leuven, Kortrijik, 8500, West Flanders, Belgium.
[3]École Polytechnique, Polytechnic Institute of Paris, Paris, 91120, Palaiseau, France.
[4]Department of Control, Automation and Computing Engineering, Federal University of Santa Catarina, Blumenau, 89036-002, SC, Brazil.
[5]Institute of Mathematical and Computing Sciences, University of São Paulo, São Carlos, 13566-590, SP, Brazil.

*Corresponding author(s). E-mail(s): elainegatto@estudante.ufscar.br;
Contributing authors: felipekenji.nakano@kuleuven.be; jesse.read@polytechnique.edu; mauri.ferrandin@ufsc.br; cerri@icmc.usp.br; celine.vens@kuleuven.be;

Abstract

Multi-label classification is a type of supervised machine learning that can simultaneously assign multiple labels to an instance. To solve this task, some methods divide the original problem into several sub-problems (local approach), others learn all labels at once (global approach), and others combine several classifiers (ensemble approach). Regardless of the approach used, exploring and learning label correlations is important to improve the classifier predictions. Ensemble of Classifier Chains (ECC) is a well-known multi-label method that considers label correlations and can achieve good overall performance on several multi-label datasets and evaluation measures. However, one of the challenges when working with ECC is the high dimensionality of the label space, which can impose limitations for fully-cascaded chains as the complexity increases regarding feature space expansion. To improve classifier chains, we propose a method to chain disjoint correlated label clusters obtained by applying a partition method in the label space. During the training phase, the ground truth labels of each cluster are used as new features for all of the following clusters. During the test phase, the predicted

labels of clusters are used as new features for all the following clusters. Our proposal, called Label Cluster Chains for Multi-Label Classification (LCC-ML), uses multi-label Random Forests as base classifiers in each cluster, combining their predictions to obtain a final multi-label classification. Our proposal obtained better results compared to the original ECC. This shows that learning and chaining disjoint correlated label clusters can better explore and learn label correlations.

Keywords: Multi-Label Classification, Multi-Label Learning, Classifier Chains

1 Introduction

Multi-label classification (MLC) is a type of supervised machine learning that allows instances to be categorized into two or more labels simultaneously [1, 2]. For example, a song can be classified into different genres at the same time [3]. This characteristic is present in various real-world domains such as text categorization [4], sentiment analysis [5], bioinformatics [6], real-time data stream mining [7], and video sports games [8].

Two main approaches to solving MLC problems exist: local (Lo) and global (G). The local approach divides the problem into sub-problems, while the global approach develops new algorithms or modifies existing ones to deal with the original problem. By dividing the data, the local approach may disregard the correlation between labels, which is considered a natural characteristic of multi-label data. On the other hand, the global approach may miss some local correlation information, as it simultaneously deals with all labels of the problem. Several studies have discussed these approaches, including Bogatinovski et al. [9], Weng et al. [10], and Endut et al. [1].

Label Correlations (or relationships between labels) are commonly found in multi-label data [11], i.e., the labels are often not independent of each other. Ignoring these correlations can lead to suboptimal performances, as the classifier may treat labels as independent when they are actually related. Taking correlations into account allows the model to exploit dependencies between labels, improving prediction accuracy. Several studies [12–16] have demonstrated that exploring these label correlations can enhance the predictive power of classifiers.

To illustrate how learning label correlations can enhance predictive performance, consider the image presented in Figure 1. Assume that during model training, a strong correlation was identified between the labels `mountain` and `beach`. In Figure 1, although the label `beach` is present in the image, it is less predominant compared to the label `mountain`. The label `mountain` is more prominent and thus easier to predict. Despite the difficulty in predicting `beach` on its own due to its lower prominence, the presence of the label `mountain` significantly increases the likelihood of predicting `beach` correctly. This is because the correlation indicates that when `mountain` is present, there is a higher probability that `beach` should also be present.

By integrating label correlations into the model, we enhance its predictive power by effectively using these relationships. For example, recognizing that `beach` often co-occurs with `mountain` allows the model to use the prediction of `mountain` as a

Fig. 1: Illustrative example of an image with correlated labels to explain difficult-to-predict labels and label correlations learning. Source: pixabay free image bank (https://pixabay.com/pt/)

strong indicator for `beach`. This ability to leverage label correlations can be effectively supported by similarity measures. Jaccard index similarity, for example, quantifies the degree of co-occurrence between label pairs by calculating the intersection ratio to the union of their occurrence sets. Specifically, if the similarity between `beach` and `mountain` is high, it indicates a frequent co-occurrence of these labels in the dataset. This measure helps identify which labels are strongly correlated and can guide the model in making informed predictions [12, 17, 18]. The model can systematically find and explore label correlations by incorporating similarity measures. This approach can enhance the model's ability to predict less frequent labels, such as `beach`, by leveraging the presence of more frequent labels, such as `mountain`. Similarity measures not only help in modeling the label correlations but also in improving the overall predictive performance of the classifier by making the best use of the identified correlations. Therefore, integrating label correlations helps enhance the model's overall performance by making more accurate predictions for labels that are less frequent or harder to predict [19–21].

Classifier Chains (CC) is a widely used approach that aims to capture these label correlations by breaking the original problem into multiple binary problems. The basic idea is to train a separate classifier for each label and then use the predictions generated by these classifiers as new features in the subsequent classifiers. The classifiers are linked in a chain, facilitating the learning of label correlations during the training process [22, 23]. However, the order of classifiers can affect its overall performance due to possible error propagation, and building all classifiers in the chain may become impractical for many labels due to increased computational complexity [2, 11, 24–26].

Ensemble of Classifier Chains (ECC) extends CC to use multiple chains of classifiers instead of a single chain. Each chain can be trained differently, and the combined results from multiple chains are used to make the final prediction. The CC

ensemble technique helps stabilize performance, improve robustness, and reduce error propagation.

Error propagation is related to the problem of chain order, which is a significant challenge in CC and was one of the reasons for developing ECC - it can influence the model's performance. If an initial classifier makes an error, that error can be propagated along the chain, affecting subsequent classifiers. The choice of label order in the chain is crucial because different orders can result in different levels of error propagation. ECC, therefore, uses multiple chains of classifiers, each with a different order, allowing the model to explore different label orders and combine the results of several chains, mitigating the negative impact of a specific order.

In some studies, attempts have been made to minimize or solve problems related to the chain order. For example, in a study by Abdullahi et al. [27], Particle Swarm Optimization was combined with a Genetic Algorithm to find an optimized label order. Similarly, Garcia et al. [28] calculated correlations between pairs of labels using imprecise probabilities and used them to determine the label order in the chain. Mishra et al. [29] suggested using multiple shorter partial chains or chains of limited length instead of a single full chain to minimize the dimensionality and error propagation.

Here you need to have a paragraph stating that ECC is still the state-of-the-art and thats why we are focusing on improving it.

Although ECC involves the construction of several chains of classifiers, as in CC, it can be more efficient and scalable than a single long chain. From this point of view, ECC can be even more scalable if the chains are constructed to deal with subgroups (clusters) of the labels. As for label correlations, the use of multiple chains adds diversity, allowing to capture a richer representation of the correlations between labels [11, 23, 25, 26].

ECC has also been found to perform well in large multi-label datasets [30], deal with label correlations, and generalize well beyond seen label sets, which is a positive aspect that can be widely explored [9, 27]. Therefore, it is a promising approach for improving the performance of multi-label classification problems.

In this paper, we present a method that takes advantage of ECC: i) tackles high dimensionality by partitioning the label space into disjoint clusters of correlated labels, where labels within each cluster are correlated with each other but are separated from other clusters. This approach allows us to model label correlations effectively within each cluster while treating each cluster as an independent unit in the chain; ii) address label correlations by incorporating them into the clustering process; iii) generate and chain these clusters in the order they are created, ensuring that each cluster reflects label correlations accurately.

We hypothesize that clustering the multi-label space improves performance by breaking the long chain of classifiers seen in ECC, which benefits datasets with high dimensionality. By learning and chaining disjoint correlated label clusters based on their correlations, we aim to enhance the classifier's prediction power. We tested this approach using 14 multi-label benchmark datasets with over 100 labels. Our findings suggest that learning and chaining disjoint correlated clusters is a promising technique that can significantly improve prediction performance.

The structure of this paper comprises the following sections: Section 2 covers related works; Section 3 explains our proposal; Section 4 details how the experiments are conducted; Section 5 presents and discusses the results; and finally, Section 6 concludes with our final thoughts and future work.

2 Related Work

Classifier Chains (CC) is a widely used method for solving problems related to multi-label classification. It is considered one of the state-of-the-art algorithms in this field [22, 23]. The method is based on Binary Relevance (BR) [31], which involves learning one model for each class. In practice, this method consists of connecting the output of one classifier to another, forming a chain. There are many variations of CC, but the basic rule is to use ground truth labels as new features for the next classifier during the training phase. During the testing phase, predicted labels are passed forward as new features for the next classifier. This way, CC can learn label correlations and overcome the limitations of Binary Relevance [22, 25].

In classifier chains, the order in which the chain is constructed has a direct impact on the final result, as classification errors are propagated along the chain. To address this problem, the Ensemble of Classifier Chains (ECC) was proposed. ECC randomly builds individual CC based on a random chain order, a random selection of instances with replacement, and obtains a set of predictions for each label, generating a ranking [22–24, 26, 32]. The final classification is then decided by majority voting. Various experiments and results have shown that ECC is one of the best algorithms for multi-label classification [24].

It is important to consider the limitations of both CC and ECC, such as increased dimensionality and high execution time, as well as the potential advantages they offer, like label correlation consideration, generalization on unknown label sets, and diversity introduction through the random order of labels in an ensemble [9, 24, 27, 29, 33]. To minimize the high dimensionality problem, one can use chains of limited length, while partial chains can help reduce error propagation [29]. Additionally, some studies propose new ways to define the order of labels, such as defining the chain order based on label correlations [28], considering it a linear ordering problem [29], or using random decision trees for dynamic order selection [34].

The literature regarding the use and applications of chains is vast, covering from proposals to solve the methodology's own problems, to solutions to problems in specific domains [23]. While it is not practical to cover all of them in this paper, some examples can be mentioned. Therefore, in this paper, we will focus on the studies that cluster or partition the label space.

The chaining concept has been applied to solve problems with missing values in data [35], to boost the performance of stacking methods [36], to aid in the identification of anti-inflammatory peptides [33], to dynamically choose the order of prediction of labels in a chain [34], to predict the patient punctuality and turnaround time in outpatient primary care settings [37], and so on.

In a previous work of ours [12], we introduced a method called Hybrid Partitions for Multi-label Classification (HPML), which aims to solve the multi-label problem by

creating partitions that sit between the classical global and local ones. HPML identifies hybrid partitions by using the entire label space to model label correlations, employing similarity metrics such as the Jaccard Index [38] to assess the similarity between label pairs (label correlations). These similarity levels are used by an Agglomerative Hierarchical Clustering Algorithm [39, 40] to first merge the clusters into a dendrogram, then cut this dendrogram to generate the hybrid partitions, and finally validate all of them to select the best one. Then, for each cluster within each partition, we induce a classifier. For example, consider a scenario where the selected hybrid partition is composed of three clusters: $P_{best} = \{C_1, C_2, C_3\}$. The clusters are defined as follows: $C_1 = \{L_1, L_4, L_6\}$, $C_2 = \{L_2, L_5\}$, and $C_3 = \{L_3\}$. In this case, we train a multi-label classifier for C_1 and another for C_2, while a local classifier is trained for C_3. The predictions from each classifier are then combined to obtain the final prediction: $\hat{P}_{best} = \{\hat{L}_1, \hat{L}_2, \hat{L}_3, \hat{L}_4, \hat{L}_5\}$, where the hat symbol indicates predicted labels. In this paper, we used a similar approach by applying chains in the disjoint correlated label clusters. An extension of Gatto et. al. [12], was proposed using community detection methods instead of clustering [19]. The results were promising in both studies.

A similar work called DTI-MLCD (drug-target interactions with multi-label learning and community detection method) is presented by Chu et al. [41] in the pharmaceutical area. The goal is to predict new targets for existing drugs and new therapies for previously identified target attributes. Using weighted label co-occurrence graphs and community detection methods, DTI-MLCD divides the label space into multiple subspaces, applying multi-label classifiers to each and combining the final predictions.

Another paper on cluster labels is presented in [42], which proposes an alternative approach to random partitioning of the label space and is data-oriented. The main goal of the mentioned paper is to evaluate how partitioning the label space using data-driven approaches can improve random partitioning in multi-label classification. The authors first use Ra*k*el to generate 250 random partitions of the data and, from these partitions, model co-occurrence graphs, apply community detection methods, and then perform classification. This study was extended in [43] and [44].

Huang et al. [45] also examined the use of the chaining strategy in combination with clustering. They proposed a framework called Group Classifier Chains (GCC) that utilizes local correlations. GCC clusters the entire training set using the chaining concept during the training phase. This means the original feature space is expanded with the label space, and clustering is applied. Once the clustering is complete, a Label Dependency Graph is modeled for each cluster, and classifier chains are applied to each graph. However, chaining is not used for new predictions.

This section presented some work related to label partitioning or clustering, showing that researchers have invested in these topics and also in modeling label correlations. However, these works (except our previous work) did not explore correlations in the way proposed in this paper: an approach that generates several partitions of labels, chooses one among them that is considered the most appropriate, and then applies chains of classifiers to that partition.

3 Label Cluster Chains Multi-Label Classifier

To find disjoint correlated label clusters, we have developed a methodology called **L**abel **C**lusters **C**hains **M**ulti-**L**abel Classifier (LCC-ML). This proposal is based on our previous work [12]. As mentioned, "disjoint correlated label clusters" refer to clusters of correlated labels that do not overlap or share any common elements (in this case, the labels from the label space). For example, if we have a partition composed of three clusters, $C_1 = \{L_1, L_4, L_6\}$, $C_2 = \{L_2, L_5\}$, and $C_3 = \{L_3\}$, they are disjoint because no label is shared between the clusters. The key feature of LCC-ML is its chaining strategy to build chains of these disjoint correlated label clusters. Our methodology is presented in Figure 2. The proposed chaining strategy is specified in step 6 and will be further detailed presented in Figures 5 and 6.

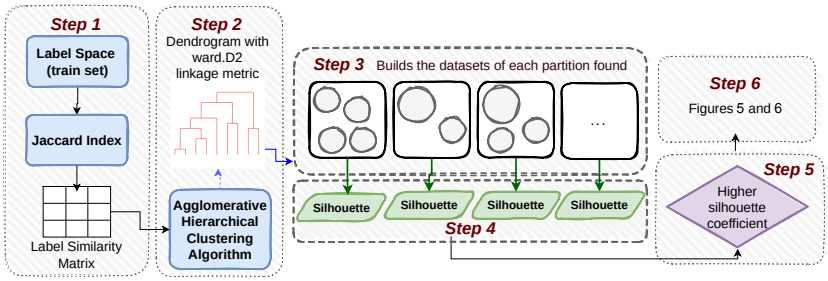


Fig. 2: LCC-ML Overview.

Our proposal's first step is to create a model for label correlations based on a similarity measure. The aim is to cluster the correlated labels into disjoint clusters. Our motivation for learning these disjoint correlated label cluster is to identify clusters of high-order correlations, which allows for easier learning of classifiers. Naturally, the labels within a cluster should be more similar to each other, while labels from different clusters should have less similarity.

The Jaccard Index Similarity Measure [38] is used to model the label correlations, taking into account only the label space from the training set. As mentioned in the introduction, label correlations can be understood as the relationships between labels. These relationships can be measured in several ways, including optimization, complex networks, label co-occurrence graphs, similarity measures, or other techniques as explored in numerous works in the literature [16–18, 46–56].

The Jaccard Index is a specific way to measure the similarity between two labels based on their co-occurrence. Although it does not directly compute the statistical correlation, i.e., it is not a measure of correlation in the classical statistical sense (like the Pearson or Spearman correlation coefficient), it provides a practical and straightforward metric for assessing the similarity between labels in a label space, being an indicator of similarity that can reflect the strength of the relationships between labels. By computing the Jaccard Index, it is possible to create a similarity matrix that reflects the relationship between pairs of labels. This matrix can then be used

to explore and model label correlations in a way that is computationally efficient and relatively simple to interpret.

The relationship between the Jaccard Index and correlation can be understood as follows: i) the Jaccard Index measures the proportion of co-occurrences between two labels compared to the total union of occurrences. It is useful for assessing the frequency with which two labels appear together relative to all possible occurrences. A higher index (value) indicates that the labels tend to appear together more frequently; ii) typically, the correlation measures the strength and direction of a linear relationship between two variables. A correlation can be, for example, positive (both variables increase or decrease together) or negative (one variable increases while the other decreases).

The Jaccard Index can serve as a practical approximation of correlation in contexts of categorical or discrete data, where the presence or absence of labels is relevant. When computing the Jaccard Index for pairs of labels, what one is essentially doing is quantifying a form of correlation based on co-occurrence. In many cases, and specifically in our approach, this is sufficient to understand and model the relationships between labels, i.e., the label correlations.

The Jaccard index computation produces a label similarity matrix that shows the similarity levels between each pair of labels. Equation 1 is used to compute the Jaccard index, which takes into account the number of times two different labels L_i and L_j occur together, as well as the number of times they never occur together. The variables a, b, c, and d respectively refer to the number of times two different labels occur together, the number of times label L_i occurs alone, the number of times label L_j occurs alone, and the number of times neither label occurs.

$$Jaccard = \frac{a}{a+b+c} \quad (1) \qquad d(Jaccard) = (1 - Jaccard) \quad (2)$$

$$\begin{array}{ll} a = \sum |(L_i == 1) \wedge (L_j == 1)| & b = \sum |(L_i == 0) \wedge (L_j == 1)| \\ c = \sum |(L_i == 1) \wedge (L_j == 0)| & d = \sum |(L_i == 0) \wedge (L_j == 0)| \end{array} \quad (3)$$

Equation 3 shows how each of these variable are computed and then used in Jaccard Equation. As a result of this calculation, we obtain a similarity matrix $M = L \times L$[1]. For example, if the label space is composed of 10 labels, then the result will be a 10 X 10 matrix with the similarity level computed for each pair of labels. The next step is to transform the similarity matrix into a dissimilarity matrix (Equation 2) and pass it as input to the ***A**gglomerative **H**ierarchical **C**lustering **A**lgorithm* (AHCA) [39, 40] to organize the labels into nested clusters - generating a dendrogram that represents the hierarchical cluster structure.

The AHCA algorithm[2] uses an agglomerative approach to hierarchical clustering, starting with each label as a separate cluster (local approach) and then merging them in consecutive steps until all labels belong to a single cluster (global approach)[40]. During the process of agglomeration, clusters are combined together based on certain linkage metrics that compute the distance between two clusters. In our proposal, we have utilized the Ward.D2 linkage metric. This particular metric was chosen because

[1] L = total number of labels in the label space of the multi-label dataset
[2] We use hclust R package: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/hclust

it leads to the formation of compact spherical clusters and is based on an objective function that determines the optimal value for selecting the pair of clusters that will be merged at each iteration [57–59].

Hierarchical clustering is a method that produces a dendrogram that represents the hierarchical cluster structure, i.e., represents the variance and distances between objects (in our case, labels) [39]. This structure can be partitioned by cutting the dendrogram at different levels in which we obtain distinct partitions of labels (disjoint label clusters). Considering this, we force the cuts in the dendrogram so that we obtain the number of partitions equal to the number of labels: $P = L$.

To illustrate this, consider Figures 3 and 4, where we have 14 labels (called Class1, Class2, and so on). In Figure 4, the blue lines are exactly where a merge occurs, indicating possible cuts for the dendrogram to generate partitions. Also, two merges at the same level are highlighted with a circle: Class1/Class2 with a value of 0.49 and Class5/Class6 with the same value. It is possible to generate 13 partitions (13 lines), but we forced 14 cuts, which generated the partitions in Table 1. Labels Class1/Class2 are always together in C_1 for partitions from P_1 to P_{10}, and are split into different clusters from P_{11} to P_{14}. A similar situation occurs with Class5/Class6: from P_1 to P_9 they are always together (C_1, C_2, C_3, C_4), and split from P_{10} to P_{14}. We can also note that Class1/Class2 and Class5/Class6 are together in the same cluster, C_1, from P_1 to P_4.

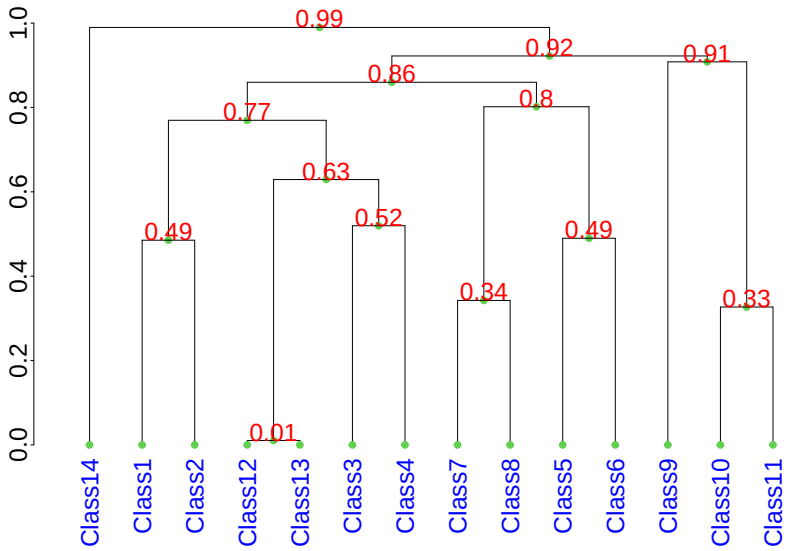


Fig. 3: Illustrative dendrogram

While a perfectly balanced dendrogram might suggest a different number of partitions, our approach enforces a cut that guarantees the number of partitions is equal

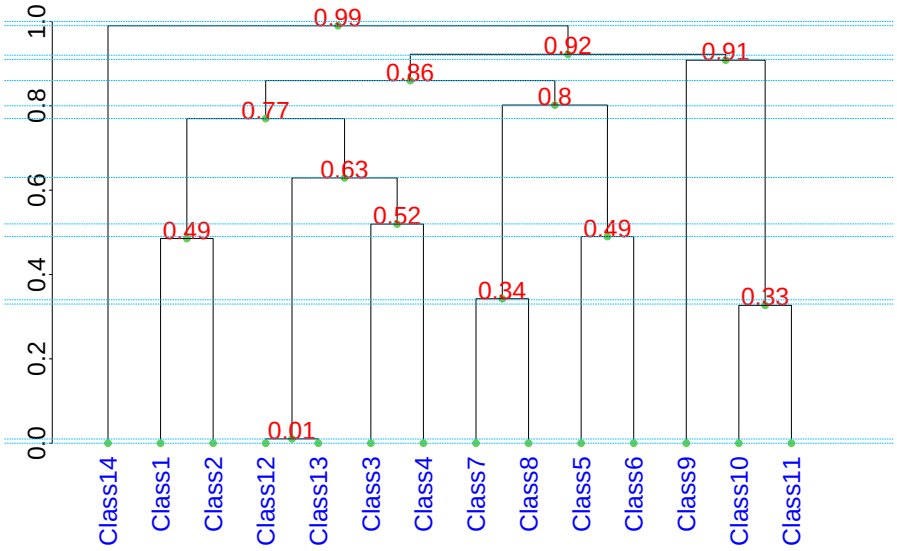


Fig. 4: Illustrative dendrogram with blue lines to illustrate the possible cuts that generates different partitions.

Table 1: Generated Partitions with forced cuts

Labels	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}
Class 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Class 2	1	1	1	1	1	1	1	1	1	1	2	2	2	2
Class 3	1	1	1	1	1	1	2	2	2	2	3	3	3	3
Class 4	1	1	1	1	1	1	2	2	3	3	4	4	4	4
Class 5	1	1	1	1	2	2	3	3	4	4	5	5	5	5
Class 6	1	1	1	1	2	2	3	3	4	5	6	6	6	6
Class 7	1	1	1	1	2	3	4	4	5	6	7	7	7	7
Class 8	1	1	1	1	2	3	4	4	5	6	7	8	8	8
Class 9	1	1	2	2	3	4	5	5	6	7	8	9	9	9
Class 10	1	1	2	3	4	5	6	6	7	8	9	10	10	10
Class 11	1	1	2	3	4	5	6	6	7	8	9	10	11	11
Class 12	1	1	1	1	1	1	2	7	8	9	10	11	12	12
Class 13	1	1	1	1	1	1	2	7	8	9	10	11	12	13
Class 14	1	2	3	4	5	6	7	8	9	10	11	12	13	14

to the number of labels. When simultaneous merges[3] occur at the same height, and a specific number of clusters is required, both merges are included to maintain the desired cluster count. However, the processing order of these merges is determined by the data's inherent ordering or the distance matrix indexing, ensuring that the final result is a number of partitions exactly matching the number of labels while adhering to the hierarchical structure imposed by the dendrogram [60–64].

[3] https://stat.ethz.ch/R-manual/R-devel/library/stats/html/cutree.html

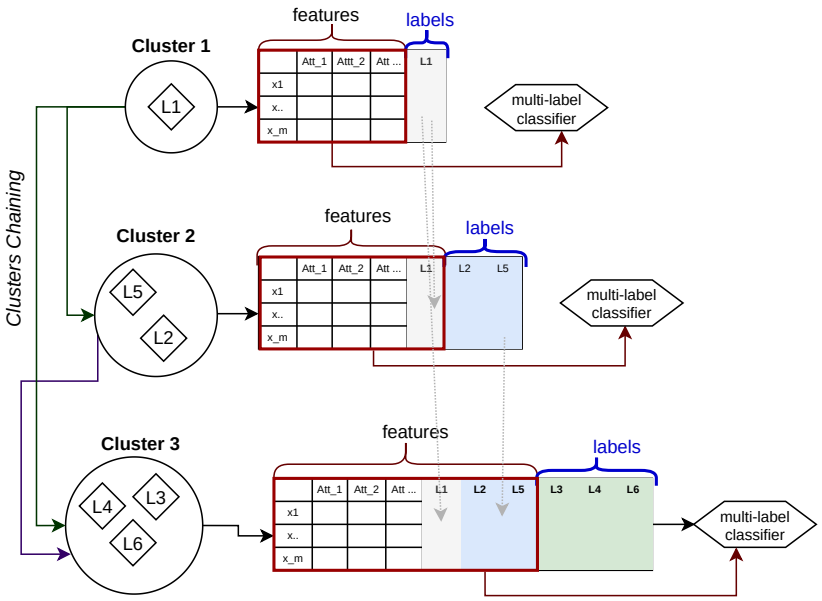


Fig. 5: Illustration of the LCC-ML training phase.

To determine the best partition among all the generated ones, in this paper, we use the silhouette coefficient [65, 66][4], which is a method to measure clustering quality. In our case, the quality of a cluster is based on the proximity between the labels within that cluster and the distance between these labels and their nearest cluster. Thus, the silhouette computation is done using the label space and not the input space. After obtaining the label partitions from the dendrogram, the partition with the highest silhouette coefficient (Figure 2, Step-5) is selected to train classifiers and make predictions for new instances (Figure 2, Step-6).

The proposed training and test procedures are illustrated in Figures 5 and 6. During training, the ground truth labels of the previous clusters are used as new features to train the classifiers for the next cluster in the chain. To obtain new predictions for a given cluster, we use the predictions on previous clusters as features. Each cluster's label predictions are modeled using Random Forests, and the final classification is obtained by combining the predictions for all clusters. Unlike classifier chains, where the chain order is randomly defined or chosen via a heuristic, the order of the clusters in our proposal is determined through cuts in a dendrogram generated by hierarchical clustering using the Ward.D2 algorithm [63]. As already mentioned, the dendrogram represents the variance and distances between labels, and by making cuts at different heights, we obtain distinct partitions of labels, which subsequently define the chain order. These clusters are ordered and chained according to the sequence in which merges occur in the dendrogram. The order of these merges is inherently tied to the

[4] https://www.rdocumentation.org/packages/cluster/versions/2.1.6/topics/silhouette

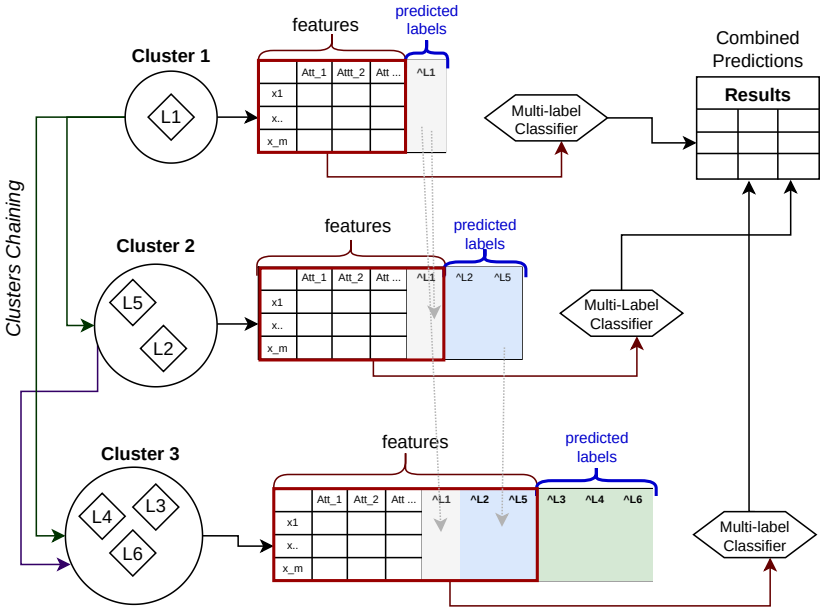


Fig. 6: Illustration of the LCC-ML predicting phase.

construction of the dendrogram, which is influenced by the data's structure and the indexing of distances within the hierarchical clustering process.

4 Experimental Set-Up

We have assessed our method using 14 multi-label datasets that have over 100 labels and belong to three different application domains. Table 2 provides a summary of their characteristics, including the dataset name, domain (D), the total number of instances (I), features (F), and labels (L). Other important characteristics include Cardinality (Card), which is the average number of labels for each instance; Density (Dens), which is the Cardinality normalized by the number of labels [2, 67]; Scumble (Score ofConcUrrence among iMBalanced LabEls) [68], that is a metric designed to quantify the level of imbalance among labels in a multi-label dataset, indicating how frequently labels with different imbalance levels co-occur within the data; TCS (Theoretical Complexity Score) [69], which calculates how hard it is to learn a predictive model from the dataset; and finally, ULD (Unconditional Label Dependency) [70], which is the average correlation of labels weighted by the number of common instances. We also computed the average (Av.), the standard deviation (S.D), and the minimum and maximum values for the selected datasets.

The source code and all necessary materials to replicate our experiments are available in our GitHub repository[5]. We conducted our experiments using a 10-fold

[5] https://github.com/cissagatto/LCC-ML

Table 2: Characteristics of the multi-label datasets used in the experiments.

Name	D	I	L	F	Card	Dens	Scumble	TCS	ULD
bibtex	Text	7395	159	1836	2.402	0.015	0.094	20.541	0.147
cal500	Music	502	174	68	26.044	0.150	0.337	15.597	0.138
corel16k001	Image	13766	153	500	2.859	0.019	0.273	19.722	0.135
corel16k003	Image	13760	154	500	2.829	0.018	0.285	19.730	0.126
corel16k004	Image	13837	162	500	2.842	0.017	0.277	19.791	0.138
corel16k005	Image	13847	160	500	2.858	0.018	0.285	19.813	0.130
corel16k006	Image	13859	162	500	2.885	0.018	0.290	19.821	0.138
corel16k007	Image	13915	174	500	2.886	0.017	0.282	19.922	0.136
corel16k008	Image	13864	168	500	2.883	0.017	0.290	19.846	0.141
corel16k009	Image	13884	173	500	2.930	0.017	0.298	19.919	0.134
corel16k010	Image	13618	144	500	2.815	0.020	0.279	19.638	0.128
rcv1sub1	Text	6000	101	47236	2.880	0.028	0.224	22.313	0.191
rcv1sub2	Text	6000	101	47236	2.634	0.026	0.209	22.238	0.224
stackex che.	Text	6961	175	540	2.109	0.012	0.187	19.473	0.053
	AV.	10800.57	154.29	7244.00	4.418	0.028	0.258	19.883	0.140
	S.D.	4475.82	24.35	16947.29	6.229	0.035	0.061	1.533	0.037
	Max.	13915.00	175.00	47236.00	26.044	0.150	0.337	22.313	0.224
	Min.	502.00	101.00	68.00	2.109	0.012	0.094	15.597	0.053

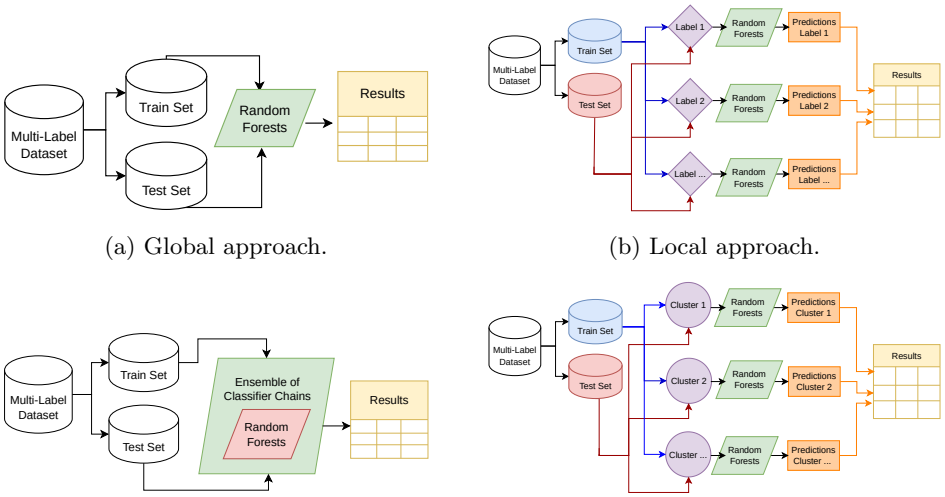
cross-validation strategy and an iterative stratification algorithm designed for multi-label data [71]. Experiments were run in a container (Apptainer[6]) within Linux clusters. To evaluate our proposed method, LCC-ML, we compared it with other methods, which are illustrated in Figure 7 and described below:

- **ECC**: Figure 7c. Is the ensemble version of classifier chains [22];
- **HPML**: Figure 7d. Our previous proposal [12];
- **Local**: Figure 7b. Each label is treated separately;
- **Global**: Figure 7a. All labels are together in one cluster.

We induced random forests in all methods with 200 trees, and the ECC consisted of 10 randomly ordered chains. We evaluated our results using four different multi-label evaluation measures. Since we used random forests as base classifiers and their outputs can be interpreted as probabilities, we computed the area under Precision-Recall (PR) Curves and Receiver Operating Characteristic (ROC) curves: AUPRC-Macro, AUPRC-Micro, ROC-AUC-Macro, and ROC-AUC-Micro [72].

Finally, to determine the significance of the results, we conducted the Friedman statistical test followed by the post-hoc Nemenyi test with a significance level of 0.05. These non-parametric tests were chosen as they do not make assumptions about the distribution of the data. We used R version 4.3.1 and the package SCMAMP version 0.3.2 to perform the statistical tests.

[6] https://github.com/apptainer/apptainer



(a) Global approach.

(b) Local approach.

(c) Ensemble of CC.

(d) HPML.

Fig. 7: Details of the baseline methods used in our experiments.

5 Results and Discussion

Table 3 displays the results of our experiments for all methods and multi-label evaluation measures used. The best results are highlighted in boldface. The table also shows the number of times each method was the best (Total Best) or worst (Total worst) across all datasets.

On the average of the 14 datasets (see average line in Table 3), we found that LCC-ML was the best in terms of AUPRC-Macro, ROC-AUC-Micro, and ROC-AUC-Macro measures. However, the Global method showed the best overall performance in AUPRC-Micro. On the other hand, the Local method obtained the worst results in AUPRC-Micro and AUPRC-Macro, while HPML had the worst results in ROC-AUC-Micro and ROC-AUC-Macro.

From all the comparisons provided, we observed that LCC-ML has outperformed ECC in several cases. This confirms our hypothesis that using chaining in the clusters in the order created by the hierarchical algorithm can solve the long chain and deal with the high dimensionality of the label space.

Although the predictive results were promising, the statistical tests revealed a different outcome. The critical distances can be seen in Figures 8, 9, 10, and 11, and the p-values and hypotheses for each measure are provided in Table 4. Based on the critical distance plots, where connected lines show where there are no significant statistical differences, it can be concluded that there is no statistically significant difference between the Local and HPML methods in all four measurements. Similarly, the four measures have no statistically significant differences between the LCC-ML, ECC, and

Table 3: Experimental results for all methods and multi-label measures

	AUPRC-MICRO					AUPRC-MACRO				
Datasets	**G**	**Lo**	**ECC**	**HPML**	**LCC-ML**	**G**	**Lo**	**ECC**	**HPML**	**LCC-ML**
bibtex	**0,4987**	0,4711	0,4867	0,4737	0,4891	**0,4190**	0,3905	0,4150	0,3939	0,4174
cal500	**0,4785**	0,4584	0,4635	0,4769	0,4775	0,2449	0,2369	0,2399	0,2463	**0,2530**
corel16k001	0,2122	0,2026	0,2124	0,2024	**0,2126**	0,1099	0,1052	0,1158	0,1051	**0,1158**
corel16k003	**0,2179**	0,2085	0,2167	0,2093	0,2170	0,1160	0,1104	**0,1214**	0,1099	0,1214
corel16k004	**0,2149**	0,2051	0,2131	0,2050	0,2122	0,1274	0,1243	0,1346	0,1238	**0,1347**
corel16k005	0,2100	0,2020	**0,2104**	0,2019	0,2089	0,1030	0,1016	**0,1122**	0,1023	0,1108
corel16k006	**0,2143**	0,2047	0,2118	0,2044	0,2115	0,1270	0,1232	**0,1338**	0,1232	0,1334
corel16k007	**0,2100**	0,2020	0,2086	0,2017	0,2099	0,1170	0,1108	0,1203	0,1098	**0,1214**
corel16k008	0,2128	0,2041	**0,2138**	0,2038	0,2133	0,1158	0,1091	**0,1187**	0,1090	0,1187
corel16k009	**0,2028**	0,1924	0,2017	0,1920	0,2004	0,1007	0,0950	**0,1049**	0,0946	0,1047
corel16k010	**0,2235**	0,2150	0,2206	0,2150	0,2207	0,1246	0,1212	**0,1329**	0,1224	0,1305
rcv1sub1	0,4810	0,4792	0,4859	0,4810	**0,4872**	0,3908	0,3630	0,3879	0,3692	**0,3947**
rcv1sub2	0,4660	0,4476	0,4644	0,4476	**0,4675**	**0,3680**	0,3252	0,3557	0,3248	0,3559
stackex_chem	**0,2704**	0,2493	0,2605	0,2574	0,2671	0,1753	0,1541	0,1668	0,1662	**0,1767**
Average	0,2938	0,2816	0,2907	0,2837	0,2925	0,1885	0,1765	0,1900	0,1786	0,1921
S.D.	0,1241	0,1207	0,1219	0,1233	0,1243	0,1171	0,1060	0,1121	0,1076	0,1139
Total Best	**9**	0	2	0	3	2	0	**6**	0	**6**
Total Worst	0	6	0	**8**	0	0	6	0	**8**	0

	ROC-AUC-MICRO					ROC-AUC-MACRO				
Datasets	**G**	**Lo**	**ECC**	**HPML**	**LCC-ML**	**G**	**Lo**	**ECC**	**HPML**	**LCC-ML**
bibtex	**0,9428**	0,9301	0,9418	0,5470	0,9423	0,9224	0,9079	0,9235	0,5893	**0,9245**
cal500	**0,8211**	0,8053	0,8068	0,5888	0,8201	0,6399	0,6348	0,6365	0,6313	**0,6480**
corel16k001	0,8567	0,8480	**0,8631**	0,4893	0,8629	0,7204	0,7106	**0,7384**	0,5344	0,7383
corel16k003	0,8605	0,8534	0,8689	0,5377	**0,8698**	0,7229	0,7155	0,7432	0,5491	**0,7460**
corel16k004	0,8632	0,8559	**0,8721**	0,5017	0,8718	0,7350	0,7302	0,7584	0,5333	**0,7586**
corel16k005	0,8546	0,8487	**0,8641**	0,5548	0,8640	0,7075	0,7046	0,7336	0,5414	**0,7341**
corel16k006	0,8601	0,8541	**0,8692**	0,5084	0,8691	0,7324	0,7269	0,7540	0,5298	**0,7541**
corel16k007	0,8606	0,8544	0,8691	0,4945	**0,8693**	0,7248	0,7183	0,7451	0,5245	**0,7461**
corel16k008	0,8600	0,8523	**0,8683**	0,4834	0,8679	0,7254	0,7161	**0,7457**	0,5270	0,7454
corel16k009	0,8574	0,8491	**0,8646**	0,4992	0,8640	0,7199	0,7107	0,7392	0,5283	**0,7395**
corel16k010	0,8624	0,8542	**0,8692**	0,5538	0,8691	0,7324	0,7237	0,7514	0,5507	**0,7518**
rcv1sub1	**0,9566**	0,9486	0,9548	0,8546	0,9556	**0,9133**	0,8781	0,9103	0,8262	0,9121
rcv1sub2	**0,9543**	0,9441	0,9504	0,9441	0,9503	**0,9121**	0,8773	0,9078	0,8777	0,9074
stackex_chem	**0,8985**	0,8749	0,8893	0,5961	0,8956	0,7758	0,7463	0,7731	0,5951	**0,7815**
Average	0,8792	0,8695	0,8823	0,5824	0,8837	0,7632	0,7501	0,7757	0,5956	0,7777
S.D.	0,0420	0,0415	0,0404	0,1401	0,0389	0,0874	0,0789	0,0811	**0,1135**	0,0798
Total Best	5	0	**7**	0	2	2	0	2	0	**10**
Total Worst	0	1	0	**13**	0	0	1	0	**13**	0

Global methods. These statistical results indicate that despite the competitive performance of our method and the assumptions being confirmed, improvements are needed to statistically surpass the other methods.

Table 5 analyzes the chosen partitions according to the silhouette coefficient criterion for each dataset and each fold. The names of the datasets have been simplified for space reasons: $c.16k$ is understood as $corel16k00$. We also add a row called L with the total number of labels for each dataset. Besides, each number in each cell in the table indicates the ID of the partition, which is also the partition size.

As an example, the *bibtex* dataset has 159 labels. This means that our approach can generate 159 different partitions. One of these partitions is the local partition (partition 159, where each label is a cluster), and the other is the global partition (partition 1, where all labels form a single cluster). In between, there are 157 hybrid

partitions ($159 - 2 = 157$), ranging from 2 to 158. For example, partition 158 has 158 clusters, where one cluster contains two labels and all others contain one label. This configuration is similar to a local partition. On the other hand, a partition with 126 clusters means that there are more clusters with single labels and fewer clusters that cluster more than one label. In contrast, partitions with 2 or 3 clusters are closer to the global partition since more labels are grouped into fewer clusters. This diversity in partitions explains why performance results are competitive between LCC-ML, global, and local partitions, a situation observed in many datasets as shown in Table 5.

Table 4: Results of the Friedman and Nemenyi statistical tests

	ChiSquare	**p-value**	**Critical Difference**	**Hyphotesis**
AUPRC-Micro	39.829	4.70E-05	1.6768	Ha:Different
AUPRC-Macro	40.914	2.8e-08	1.6768	Ha:Different
ROC-AUC-Micro	44.8	4.38E-06	1.6768	Ha:Different
ROC-AUC-Macro	48.229	8.46E-07	1.6768	Ha:Different

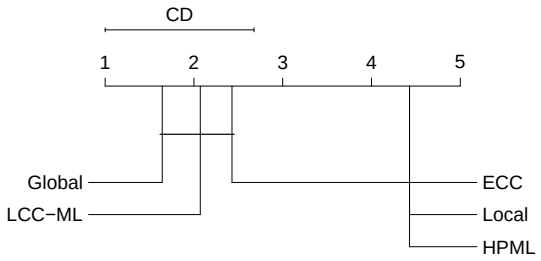


Fig. 8: Critical Distance: AUPRC-Micro

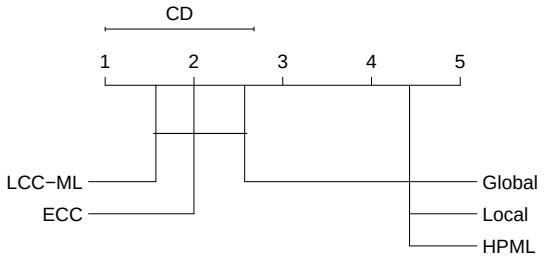


Fig. 9: Critical Distance: AUPRC-Macro

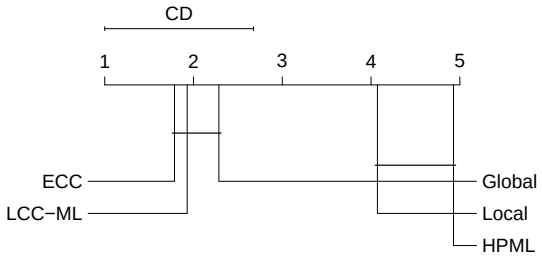


Fig. 10: Critical Distance: ROC-AUC-Micro

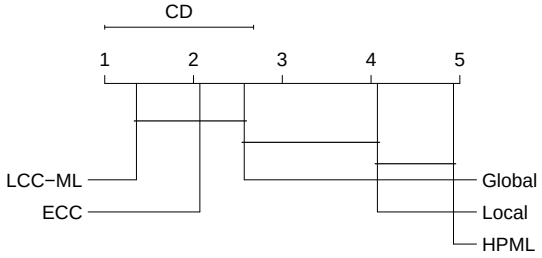


Fig. 11: Critical Distance: ROC-AUC-Macro

Table 5: Chosen Partitions. The numbers in each cell are the partition ID. For example, the number 158 is partition 158 (P158), composed of 158 clusters.

fold	bibtex	cal500	c.16k1	c.16k3	c.16k4	c.16k5	c.16k6	c.16k7	c.16k8	c.16k9	c.16k10	r1s1	r1s2	s_ch
L	159	174	153	154	162	160	162	174	168	173	144	101	101	175
1	128	2	134	4	143	141	140	149	144	147	131	100	100	3
2	124	2	135	4	140	139	139	149	148	150	131	2	100	2
3	131	2	136	4	139	144	138	149	149	150	131	100	100	2
4	126	2	135	141	141	142	138	152	144	150	131	100	100	172
5	126	2	135	141	141	144	140	151	145	148	131	100	100	174
6	131	2	133	4	140	143	137	149	145	147	131	100	100	172
7	128	173	136	145	140	140	138	146	145	149	131	100	100	174
8	125	2	136	141	143	141	137	151	144	148	131	100	100	174
9	124	2	134	140	143	142	141	151	144	146	131	100	100	174
10	126	2	132	141	143	144	138	149	144	147	131	2	100	2

We can notice a consistency in the choice of partitions in most datasets, indicating a range of partitions composed of similar clusters that are considered the highest quality according to the silhouette coefficient. The performance result is based on the average of the ten folds executed with these chosen partitions. Table 5 shows that partitions with more clusters have better clustering quality. By chaining the clusters created for each partition, allowing the transmission of correlations in the chain, the

expectation is that this chosen partition will help improve the classifier's performance and overcome ECC.

When analyzing the performance results of LLC-ML with different partitions, we observed that partitions with many clusters improve the classifier performance, especially when using cluster chaining. For example, in datasets such as *corel*16*k*001, *corel*16*k*008, *rcv*1*sub*1, and *rcv*2*sub*2, where the chosen partitions contain many clusters, LLC-ML does not achieve the best performance in ROC-AUC-Macro. However, the partitions comprised many clusters on other datasets where LLC-ML performed best. This suggests that, although the method is not the best in some specific cases, the general trend is that many clusters can contribute to the better performance of LLC-ML.

Looking at the ROC-AUC-Macro metric results, it is interesting to note that in the *rcv*1*sub*2 dataset, the same partition was tested on ten different folds, which is generally desirable to obtain consistent results across methods and datasets. However, LLC-ML was not the best in this scenario. In contrast, LLC-ML performed best in the *corel*16*k*010 dataset when the same partition was used for all folds.

This difference may be attributed to details more related to the data than to the compared methods or partitions, such as ULD and TCS, which are characteristics of multi-label datasets. We can see from Table 2 that the TCS for *rcv*1*sub*1 is 22.238, and for *corel*16*k*010 is 19.638, a difference of 2.600. Thus *rcv*1*sub*1 is more challenging to learn than *corel*16*k*010. Concerning the ULD, *corel*16*k*010 (0.128) has an unconditional label dependency value lower than *rcv*1*sub*1 (0.224), which indicates that the labels of the former are more correlated than those of the latter. This may be an indication that LCC-ML was able to better capture these correlations from the *corel*16*k*010 during the chaining, leading to better performance in the ROC-AUC-Macro measure when compared to *rcv*1*sub*1. We can conclude that this also happened for the other methods, datasets and evaluation measures.

5.1 Complexity analysis

This section presents a complexity analysis of the algorithms using the following notation: symbol C represents the total number of clusters, while P denotes the total number of partitions for a dataset; the total number of instances is denoted by I, and F refers to the total number of input features or attributes; variable T corresponds to the total number of trees and L to the total number of labels; M is used for the total number of models in an ensemble.

The total complexities of the Global and Local approaches are $O(T \times I \times F \log I)$ and $O(L \times T \times I \times F \log I)$. Training ECC with Random Forests takes $O(M \times L^2 \times T \times I \times F \times \log I)$, while testing takes $O(M \times L^2 \times T \times I \times F \log I)$. Therefore, the total complexity of ECC is $O(M \times L^2 \times T \times I \times F \log I)$.

The Jaccard Index computation takes $O(L^3)$ to iterate over all labels and compute when they occur together or alone. The agglomerative hierarchical clustering algorithm takes $O(L^3)$ since we must consider the iterations and the operations within each iteration. This complexity arises because, for each of the $L-1$ iterations, the dominant operation is to update the distance matrix, which can involve up to $O(L^2)$ comparisons and updates [73].

The Silhouette Coefficient of a data point (in our case, the data points are the labels) is calculated by comparing it with all other data points in its cluster and with points in other clusters. Therefore, the complexity of calculating the silhouette is $O(L^2)$ as we need to consider the operation of comparing pairs of data points.

We must compare the results from all partitions to select the best partition. This takes $O(P)$ because we iterate over all the silhouette results. Therefore, the validation takes $O(L^2)$, while the train/test of the chosen partition takes $O(T \times I \times F \log I)$.

The total complexity of HPML is determined by the largest of the complexities involved. In this case, the dominant complexities are $O(L^3)$ of the Jaccard Index and the Hierarchical Clustering Algorithm and $O(T \times I \times F \log I)$ of the Training/Testing of the chosen partition. Therefore: $O(L^3) + O(T \times I \times F \log I)$.

For training and testing LCC-ML, it is necessary to consider training each classifier on each cluster. There are C clusters, and each cluster C_i has L labels; the training complexity of a chain can be approached as i) for each cluster C_i where its labels are used as features for cluster C_{i+1}, the complexity is $O(T \times I \times F \log I)$; and ii) for C clusters, the total training complexity is $O(C \times T \times I \times F \log I)$.

In the testing phase, the predicted labels of each cluster are used as features for the next cluster in the chain. Thus i) for each cluster in the chain, the classifier that uses the predicted labels of the previous cluster as features is evaluated, therefore, $O(T \times I \times F \log I)$ for each cluster; ii) since there are C clusters, the total testing complexity is $O(C \times T \times I \times F \log I)$. Thus, the total complexity for LCC-ML is $O(L^3) + O(C \times T \times I \times F \log I)$.

We can conclude that ECC is the more complex method, while Global is the most minor complex. ECC is the most complex due to the presence of the $M \times L^2$ factor, which causes the complexity to scale significantly with the number of labels (L) and models (M). Global is the most minor complex since it does not depend on the number of labels (L) or clusters (C), resulting in relatively low complexity.

6 Conclusions

In this paper, we introduced a new method for chaining correlated label clusters called LCC-ML. It groups correlated labels into clusters based on their modeled correlations. These clusters are then used in the learning process through a chaining approach. We generated these clusters using the Agglomerative Hierarchical Clustering Algorithm, with the Ward.D2 linkage metric, in a label similarity matrix generated using Jaccard index.

We performed experiments using 14 benchmark multi-label datasets from three distinct domains, each with over 100 labels in the label space. We assessed the results using four multi-label evaluation measures. Additionally, we compared LCC-ML with ECC, HPML, and the local and global versions of Multi-label Random Forests.

The performance evaluation results indicated that for datasets with a high number of labels, breaking the long chain and learning disjoint correlated label clusters in a chaining cluster order based on label correlations can lead to effective learning. LCC-ML outperformed ECC in several cases, which is considered one of the best methods in multi-label classification. Additionally, LCC-ML overcame the Global and Local

approaches, as we hypothesized. However, LCC-ML still requires improvements, as statistical tests showed no significant differences between the compared methods.

Our next steps involve conducting further tests of the method using different datasets, multi-label classifiers, evaluation measures, and linkage metrics for the agglomerative hierarchical clustering algorithm. We could also explore other ways to model the label correlations and extend LCC-ML to hierarchical and extreme multi-label problems. Additionally, we could consider implementing complete chaining, linking the first and last clusters.

Acknowledgments.

This study was financed by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brazil, Finance Code 200371/2022-3, the Research Fund Flanders (through research projects 1235924N to FKN and CV), and the Flemish AI Research Program.

References

[1] Endut, N., Hamzah, W.M.A.F.W., Ismail, I., Yusof, M.K., Baker, Y.A., Yusoff, H.: A Systematic Literature Review on Multi-Label Classification based on Machine Learning Algorithms. TEM Journal **11**(2), 658–666 (2022) https://doi.org/10.18421/TEM112-20

[2] Herrera, F., Charte, F., Rivera, A.J., Jesus, M.J.: Multilabel Classification: Problem Analysis, Metrics and Techniques, 1st edn. Springer, ??? (2016)

[3] Sanden, C., Zhang, J.Z.: Enhancing multi-label music genre classification through ensemble techniques. In: SIGIR'11 - Proc. 34th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., pp. 705–714 (2011). https://doi.org/10.1145/2009916.2010011

[4] Chen, S., Kuang, Q., Yu, X., Li, S., Ding, R.: A multi-label classification algorithm for non-standard text. In: 2022 International Conference on Asian Language Processing (IALP), pp. 206–211 (2022). https://doi.org/10.1109/IALP57159.2022.9961273

[5] Deng, J., Ren, F.: Multi-label emotion detection via emotion-specified feature extraction and emotion correlation learning. IEEE Transactions on Affective Computing **14**(1), 475–486 (2023) https://doi.org/10.1109/TAFFC.2020.3034215

[6] Chen, S., Wang, J., Guo, M.: A new method for predicting plant proteins function based on multi label classification algorithm. In: 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 2067–2073 (2022). https://doi.org/10.1109/BIBM55620.2022.9994961

[7] Luo, F., Liu, G., Guo, W., Chen, G., Xiong, N.: Ml-kelm: A kernel extreme learning machine scheme for multi-label classification of real time data stream

in siot. IEEE Transactions on Network Science and Engineering **9**(3), 1044–1055 (2022) https://doi.org/10.1109/TNSE.2021.3073431

[8] He, J., Pao, H.-K.: Multi-modal, multi-labeled sports highlight extraction. In: 2020 International Conference on Technologies and Applications of Artificial Intelligence (TAAI), pp. 181–186 (2020). https://doi.org/10.1109/TAAI51410.2020.00041

[9] Bogatinovski, J., Todorovski, L., Džeroski, S., Kocev, D.: Comprehensive comparative study of multi-label classification methods. Expert Systems with Applications **203** (2022) https://doi.org/10.1016/j.eswa.2022.117215 arXiv:2102.07113

[10] Weng, W., Li, Y.-W., Liu, J.-H., Wu, S.-X., Chen, C.-L., Weng, W., Li, Y.W., Liu, J.H., Wu, S.X., Chen, C.L.: Multi-Label Classification Review and Opportunities. Taiwan Ubiquitous Information **6**(2) (2021)

[11] Read, J.: Scalable multi-label classification. PhD thesis, University of Waikato (2010)

[12] Gatto, E.C., Ferrandin, M., Cerri, R.: Exploring label correlations for partitioning the label space in multi-label classification. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2021). https://doi.org/10.1109/IJCNN52387.2021.9533331

[13] Glinka, K., Wosiak, A., Zakrzewska, D.: Exploiting label interdependencies in multi-label classification. In: Burduk, R., Kurzynski, M., Wozniak, M. (eds.) Progress in Computer Recognition Systems, pp. 57–66. Springer, Cham (2020)

[14] J. Barezi, E., Kwok, J.T., Rabiee, H.R.: Multi-Label learning in the independent label sub-spaces. Pattern Recognit. Lett. **97**, 8–12 (2017) https://doi.org/10.1016/j.patrec.2017.06.024

[15] Nguyen, T.T., Nguyen, T.T.T., Luong, A.V., Nguyen, Q.V.H., Liew, A.W.-C., Stantic, B.: Multi-label classification via label correlation and first order feature dependance in a data stream. Pattern Recognition **90**, 35–51 (2019) https://doi.org/10.1016/j.patcog.2019.01.007

[16] Zhu, Y., Kwok, J.T., Zhou, Z.-H.: Multi-label learning with global and local label correlation. IEEE Transactions on Knowledge and Data Engineering **30**(6), 1081–1094 (2018) https://doi.org/10.1109/TKDE.2017.2785795

[17] Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms **26**(8), 1819–1837 (2014) https://doi.org/10.1109/TKDE.2013.39

[18] Zhao, T., Zhang, Y., Pedrycz, W.: Robust multi-label classification with enhanced global and local label correlation. Mathematics **10** (2022) https://doi.org/10.3390/math10111871

[19] Gatto, E.C., Valejo, A.D.B., Ferrandin, M., Cerri, R.: Community detection for multi-label classification. In: Naldi, M.C., Bianchi, R.A.C. (eds.) Intelligent Systems, pp. 78–93. Springer, Cham (2023)

[20] Nan, G., Li, Q., Dou, R., Liu, J.: Local positive and negative correlation-based k-labelsets for multi-label classification. Neurocomputing **318**, 90–101 (2018) https://doi.org/10.1016/j.neucom.2018.08.035

[21] Yu, P., Jia, X.: Exploiting indirect linear correlation for label distribution learning. Neurocomputing **599**, 128022 (2024) https://doi.org/10.1016/j.neucom.2024.128022

[22] Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **5782 LNAI**(PART 2), 254–269 (2009) https://doi.org/10.1007/978-3-642-04174-7_17

[23] Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains: A review and perspectives. J. Artif. Intell. Res. **70**, 683–718 (2021) https://doi.org/10.1613/JAIR.1.12376

[24] Moyano, J.M., Gibaja, E.L., Cios, K.J., Ventura, S.: Review of ensembles of multi-label classifiers: Models, experimental study and prospects. Information Fusion **44**(November 2017), 33–45 (2018) https://doi.org/10.1016/j.inffus.2017.12.001

[25] Read, J., Hollmén, J.: A deep interpretation of classifier chains. In: Blockeel, H., Leeuwen, M., Vinciotti, V. (eds.) Advances in Intelligent Data Analysis XIII, pp. 251–262. Springer, Cham (2014)

[26] Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine Learning **85**(3), 333–359 (2011) https://doi.org/10.1007/s10994-011-5256-5

[27] Abdullahi, A., Samsudin, N.A., Khalid, S.K.A., Othman, Z.A.: An Improved Multi-label Classifier Chain Method for Automated Text Classification. Int. J. Adv. Comput. Sci. Appl. **12**(3), 442–449 (2021) https://doi.org/10.14569/IJACSA.2021.0120352

[28] Moral-García, S., Castellano, J.G., Mantas, C.J., Abellán, J.: A new label ordering method in Classifier Chains based on imprecise probabilities. Neurocomputing **487**, 34–45 (2022) https://doi.org/10.1016/j.neucom.2022.02.048

[29] Mishra, N.K., Singh, P.K.: Linear Ordering Problem based Classifier Chain using Genetic Algorithm for multi-label classification. Appl. Soft Comput. **117**, 108395 (2022) https://doi.org/10.1016/j.asoc.2021.108395

[30] B, S.G.: A Survey on Ensemble Multi-label Classifiers vol. 1, pp. 100–109.

Springer, ??? (2023). https://doi.org/10.1007/978-3-031-27524-1

[31] Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. Pattern Recognition **37**(9), 1757–1771 (2004) https://doi.org/10.1016/j.patcog.2004.03.009

[32] Madjarov, G., Kocev, D., Gjorgjevikj, D., Džeroski, S.: An extensive experimental comparison of methods for multi-label learning, vol. 45, pp. 3084–3104 (2012). https://doi.org/10.1016/j.patcog.2012.03.004

[33] Zhang, J., Zhang, Z., Pu, L., Tang, J., Guo, F.: AIEpred: An Ensemble Predictive Model of Classifier Chain to Identify Anti-Inflammatory Peptides. IEEE/ACM Trans. Comput. Biol. Bioinforma. **18**(5), 1831–1840 (2021) https://doi.org/10.1109/TCBB.2020.2968419

[34] Mencía, E.L., Kulessa, M., Bohlender, S., Fürnkranz, J.: Tree-based dynamic classifier chains. Machine Learning (2022) https://doi.org/10.1007/s10994-022-06162-3

[35] Antonenko, E., Read, J.: Chains of autoreplicative random forests for missing value imputation in high-dimensional datasets. To appear in: Machine Learning journal (ECML/PKDD 2024 Journal Track), 1–16 (2022)

[36] Mishra, N.K., Himthani, P.K., Singh, P.K.: StaC: Stacked chaining for multi-label classification. Expert Syst. Appl. **219**(May 2022), 119699 (2023) https://doi.org/10.1016/j.eswa.2023.119699

[37] Lekham, L.A., Wang, Y., Hey, E., Khasawneh, M.T.: Multi-label chain clustering-classification and regression predictive models for patient punctuality and turnaround time in outpatient primary care settings. IISE Transactions on Healthcare Systems Engineering **12**, 288–304 (2022) https://doi.org/10.1080/24725579.2022.2068703

[38] Jaccard, P.: The distribution of the flora in the alpine zone. New Phytologist **11**(2), 37–50 (1912)

[39] Lukasov, A.: Hierarchical agglomerative clustering procedure. Pattern Recognition **11**, 365–381 (1978)

[40] Ran, X., Xi, Y., Lu, Y., Wang, X., Lu, Z.: Comprehensive Survey on Hierarchical Clustering Algorithms and the Recent Developments vol. 222. Springer, ??? (2022). https://doi.org/10.1007/s10462-022-10366-3 . https://doi.org/10.1007/s10462-022-10366-3

[41] Chu, Y., Shan, X., Salahub, D.R., Xiong, Y., Wei, D.-Q.: Predicting drug-target interactions using multi-label learning with community detection method (dti-mlcd). bioRxiv (2020) https://doi.org/10.1101/2020.05.11.087734

https://www.biorxiv.org/content/early/2020/05/12/2020.05.11.087734.full.pdf

[42] Szymański, P., Kajdanowicz, T., Kersting, K.: How is a data-driven approach better than random choice in label space division for multi-label classification? Entropy **18**(8), 1–23 (2016) https://doi.org/10.3390/e18080282 arXiv:1606.02346

[43] Wang, X., Zhu, X., Ye, M.: Sts-nlsp: A network-based label space partition method for predicting the specificity of membrane transporter substrates using a hybrid feature of structural and semantic similarity. Frontiers in Bioengineering and Biotechnology **7**, 306 (2019) https://doi.org/10.3389/fbioe.2019.00306

[44] Wang, X., Xie, J., Tao, X., Zhang, X., Lu, Q.: Multi-label learning based on clustering optimization. Proceedings - 2019 IEEE International Conferences on Ubiquitous Computing and Communications and Data Science and Computational Intelligence and Smart Computing, Networking and Services, 224–229 (2019) https://doi.org/10.1109/IUCC/DSCI/SmartCNS.2019.00066

[45] Huang, J., Li, G., Wang, S., Zhang, W., Huang, Q.: Group sensitive Classifier Chains for multi-label classification. Proceedings - IEEE International Conference on Multimedia and Expo **2015-Augus**, 1–6 (2015) https://doi.org/10.1109/ICME.2015.7177400

[46] Dembczyński, K., Waegeman, W., Cheng, W., Hüllermeier, E.: On label dependence in multi-label classification. Mach. Learn. **88**(1-2), 5–45 (2012) https://doi.org/10.1007/s10994-012-5285-8

[47] Huang, S.-J., Zhou, Z.-H.: Multi-label learning by exploiting label correlations locally. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence. AAAI'12, pp. 949–955. AAAI Press, Toronto Ontario Canada (2012)

[48] Li, Y., Yang, L.: More correlations better performance: Fully associative networks for multi-label image classification. 2020 25th International Conference on Pattern Recognition (ICPR), 9437–9444 (2021)

[49] Shi, Z., Wen, Y., Feng, C., Zhao, H.: Drift detection for multi-label data streams based on label grouping and entropy. IEEE International Conference on Data Mining Workshops, ICDMW **2015-Janua**(January), 724–731 (2015) https://doi.org/10.1109/ICDMW.2014.92

[50] Wang, L., Ding, Z., Lee, K., Han, S., Han, J.-J., Choi, C., Fu, Y.: Generative multi-label correlation learning. ACM Transactions on Knowledge Discovery from Data (2022) https://doi.org/10.1145/3538708

[51] Weng, W., Lin, Y., Wu, S., Li, Y., Kang, Y.: Multi-label learning based on label-specific features and local pairwise label correlation. Neurocomputing **273**, 385–394 (2018) https://doi.org/10.1016/j.neucom.2017.07.044

[52] Xu, J., Tian, H., Wang, Z., Wang, Y., Kang, W., Chen, F.: Joint input and output space learning for multi-label image classification. IEEE Transactions on Multimedia **23**, 1696–1707 (2021) https://doi.org/10.1109/TMM.2020.3002185

[53] Ye, C., Wu, J., Sheng, V.S., Zhao, P., Cui, Z.: Multi-label active learning with label correlation for image classification. In: 2015 IEEE International Conference on Image Processing (ICIP), pp. 3437–3441 (2015). https://doi.org/10.1109/ICIP.2015.7351442

[54] Zhang, M.-L., Wu, L.: Lift: Multi-label learning with label-specific features. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(1), 107–120 (2015) https://doi.org/10.1109/TPAMI.2014.2339815

[55] Zhang, X.Y., Min, F., Song, G., Yu, H.: Lstc: When label-specific features meet third-order label correlations. Information Sciences **632**, 617–636 (2023) https://doi.org/10.1016/j.ins.2023.03.056

[56] Zhao, F., Ai, Q., Li, X., Wang, W., Gao, Q., Liu, Y.: Tlc-xml: Transformer with label correlation for extreme multi-label text classification. Neural Processing Letters **56** (2024) https://doi.org/10.1007/s11063-024-11460-z

[57] Murtagh, F., Legendre, P.: Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion? Journal of Classification **31**(3), 274–295 (2014) https://doi.org/10.1007/s00357-014-9161-z

[58] Theodoridis, S., Koutroumbas, K.: Clustering algorithms ii: Hierarchical algorithms. Pattern Recognition, 653–700 (2009) https://doi.org/10.1016/b978-1-59749-272-0.50015-3

[59] Ward, J.H.: Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association **58**(301), 236–244 (1963)

[60] Team, R.C.: Stats: The R Stats Package. (2020). R package version 3.6.2. https://www.rdocumentation.org/packages/stats/versions/3.6.2

[61] Team, R.C.: Stats: The R Stats Package. (2023). R package version 4.3.1, function cutree. https://stat.ethz.ch/R-manual/R-devel/library/stats/html/cutree.html

[62] Jr., J.H.W.: Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association **58**(301), 236–244 (1963) https://doi.org/10.1080/01621459.1963.10500845 https://www.tandfonline.com/doi/pdf/10.1080/01621459.1963.10500845

[63] Murtagh, F., Legendre, P.: Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion? Journal of Classification **31**, 274–295 (2014) https://doi.org/10.1007/s00357-014-9161-z

[64] Ruppert, D.: The elements of statistical learning: Data mining, inference, and prediction. Journal of the American Statistical Association **99**, 567–567 (2004) https://doi.org/10.1198/jasa.2004.s339

[65] Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**(1), 53–65 (1987) https://doi.org/10.1016/0377-0427(87)90125-7

[66] Shahapure, K.R., Nicholas, C.: Cluster quality analysis using silhouette score. In: 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA) (2020). https://doi.org/10.1109/DSAA49011.2020.00096

[67] Gibaja, E., Ventura, S.: A tutorial on multilabel learning. ACM Comput. Surv. **47**(3) (2015) https://doi.org/10.1145/2716262

[68] Charte, F., Rivera, A., Jesus, M.J., Herrera, F.: Concurrence among imbalanced labels and its influence on multilabel resampling algorithms. In: Polycarpou, M., Carvalho, A.C.P.L.F., Pan, J.-S., Woźniak, M., Quintian, H., Corchado, E. (eds.) Hybrid Artificial Intelligence Systems, pp. 110–121. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07617-1_10

[69] Charte, F., Rivera, A., Jesus, M.J., Herrera, F.: On the impact of dataset complexity and sampling strategy in multilabel classifiers performance. In: Hybrid Artificial Intelligent Systems, pp. 500–511. Springer, ??? (2016)

[70] Luaces, O., Díez, J., Barranquero, al., J.: Binary relevance efficacy for multilabel classification, 303–313 (2012) https://doi.org/10.1007/s13748-012-0030-x

[71] Sechidis, K., Tsoumakas, G., Vlahavas, I.: On the stratification of multi-label data. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **6913 LNAI**, 145–158 (2011) https://doi.org/10.1007/978-3-642-23808-6_10

[72] Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: Proceedings of the 23rd International Conference on Machine Learning. ICML '06, pp. 233–240 (2006)

[73] Bataineh, B.: Fast component density clustering in spatial databases: A novel algorithm. Information **13**(10) (2022) https://doi.org/10.3390/info13100477