Learning from multi-label data with interactivity constraints: an extensive experimental study

Noureddine-Yassine Nair-Benrekia, Pascale Kuntz, Frank Meyer

Learning from multi-label data with interactivity constraints: an extensive experimental study

Noureddine-Yassine Nair-Benrekia[a,b,*], Pascale Kuntz[b], Frank Meyer[a]

[a] *Orange Labs, Avenue Pierre Marzin, 22307 Lannion cedex France.*
[b] *Laboratoire d'Informatique de Nantes Atlantique - Site Polytech'Nantes - La Chantrerie-BP 50609, 44360 NANTES cedex France*

Abstract

Interactive classification aims at introducing user preferences in the learning process to produce individualized outcomes more adapted to each user's behaviour than the fully automatic approaches. The current interactive classification systems generally adopt a single-label classification paradigm that constrains items to span one label at a time and consequently limit the user's expressiveness while he/she interacts with data that are inherently multi-label. Moreover, the experimental evaluations are mainly subjective and closely depend on the targeted use cases and the interface characteristics. This paper presents the first extensive study of the impact of the interactivity constraints on the performances of a large set of twelve well-established multi-label learning methods. We restrict ourselves to the evaluation of the classifier predictive and time-computation performances while the number of training examples regularly increases and we focus on the beginning of the classification task where few examples are available. The classifier performances are evaluated with an experimental protocol independent of any implementation environment on a set of twelve multi-label benchmarks of various sizes from different domains. Our comparison shows that four classifiers can be distinguished for the prediction quality: RF-PCT (Random Forest of Predictive Clustering Trees, Kocev (2012)), EBR (Ensemble of Binary Relevance, (Read et al., 2011)), CLR (Calibrated Label Ranking, Fürnkranz et al. (2008)) and MLkNN (Multi-label kNN, Zhang and Zhou (2007)) with an advantage for the first two ensemble

classifiers. Moreover, only RF-PCT competes with the fastest classifiers and is therefore considered as the most promising classifier for an interactive multi-label learning system.
Keywords:
interactive learning, multi-label learning, comparative study.

1. Introduction

By integrating some user preferences in a classification process, human-centered systems aim at producing individualized outcomes more adapted to each user's behavior than the fully automatic approaches (e.g. Amershi et al. (2015); Porter et al. (2013); Amershi (2011); Lintott et al. (2008); Fails and Olsen Jr (2003); Ware et al. (2001)). When the preferences are made explicit, they can be integrated in the learning model, for instance by defining some constraints on the dataset (Wagstaff et al., 2001; Bilenko et al., 2004). Otherwise, an alternative is to let the user interact with the system which dynamically learns from his/her behavior. As recently defined by Amershi et al. (2015) « interactive machine learning is a process that involves a tight interaction loop between a human and a machine learner, where the learner iteratively takes input from the human, promptly incorporates that input, and then provides the human with output impacted by the results of the iteration ».

In a classification framework, the learning algorithm tries to quickly build a first predictive model from a restricted set of examples given by the user and it presents him/her with personalized predictions. For instance, to query a Video on Demand (*VoD*) catalogue for a good film to watch, a user defines his/her target concepts such as « Funny », « Masterpiece », and « Fairytale » and with an adapted interface he/she labels a small set of familiar films (e.g. Ice Age (« Funny »), Avatar (« Masterpiece », « Fairytale ») (see Figure 1).

*Corresponding author at Orange Labs, Avenue Pierre Marzin, 22307 Lannion cedex France. Tel.: +33 2 96 07 99 70
Email addresses: yacinenoureddine.nairbenrekia@orange.com (Noureddine-Yassine Nair-Benrekia), pascale.kuntz@univ-nantes.fr (Pascale Kuntz), franck.meyer@orange.com (Frank Meyer)

A classification algorithm tries to capture the user's preferences and to learn a predictive model which provides the user with other relevant films from the catalogue. To strengthen the predictive performance of the model, the user regularly inspects the quality of the predictions and possibly corrects the misclassified examples (i.e. relevance feedback (Stumpf et al., 2007; Salton and Buckley, 1997)).

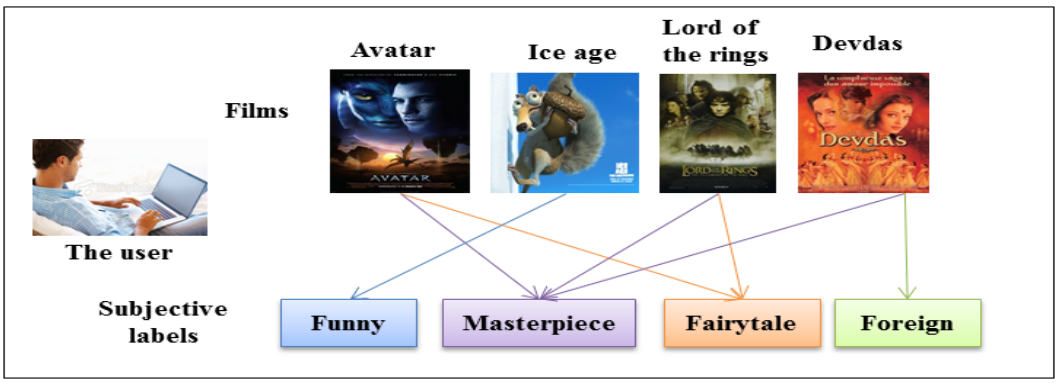


Figure 1: Interactive classification of a *VoD* catalogue (a toy example).

The increasing importance currently given to personalized contents has led to the development of several interactive classification systems for various real-world applications: e.g. image classification (Fogarty et al., 2008), file selection (Ritter and Basu, 2009), gesture classification (Fiebrink et al., 2009), document classification (Drucker et al., 2011), alarm triage (Amershi et al., 2011) and profile classification in social networks (Amershi et al., 2012). Some experimental results on the targeted domain are promising but each of these approaches adopts a single-label classification paradigm that constrains items to span one label at a time. This simplifying framework significantly limits the user's expressiveness while he/she interacts with data that are inherently multi-label.

Learning from multi-label data has received significant attention over the past few years from machine learning and its related communities (Zhang and Zhou, 2013; Madjarov et al., 2012; Sorower, 2010; Tsoumakas et al., 2010; Tsoumakas and Katakis, 2007). Initially developed for text categorization (Schapire and Singer, 2000), approaches have been extended to

diverse application domains: classification of multimedia contents including image (Boutell et al., 2004), audio (Lo et al., 2011) and video (Snoek et al., 2006), web and rule mining (Ozonat and Young, 2009; Rak et al., 2005), bioinformatics (Clare and King, 2001), tag recommendation (Katakis et al., 2008) and information retrieval (Yu et al., 2005).

Our final objective in the next future is to integrate a multi-label approach into an interactive classification system to allow users to label examples with several subjective labels of interest and consequently express complex search queries on data; *VoD* being one of our privileged application field. And the first major issue common to all developers of such a system is: which multi-label classifier should we choose? The efficiency of a real-life interactive machine learning system depends on different factors such as, in particular, the quality of the learner, the data visualization display and the interaction tools. In this paper, we restrict ourselves to the analysis of the learner behaviours without taken into account the human interface dimension and we evaluate their performances with an experimental protocol independent of any implementation environment.

More precisely, we here consider the two following major constraints: learning from few training examples in a limited time. And we study the impact of these constraints on the performances of a large set of twelve well-established algorithms from the three major families of multi-label learning methods: five problem transformation methods, two algorithm adaptation methods and five ensemble methods. We evaluate each classifier on a collection of nested training sets of increasing sizes and we focus on the beginning of the classification process where the number of examples is limited. The literature presents a wide range of measures for the evaluation of the classifier predictive performances (e.g. (Tsoumakas et al., 2010; Zhang and Zhou, 2013)) but there is no consensus on the podium of the « best » measures. We here focus on the most useful criteria in the interactive context: ranking labels by relevance, ranking examples by relevance and classifying labels. These criteria have led us to select four measures from the literature (Ranking Loss, Accuracy,

F_1-score and multi-label Balanced Error Rate BER) and to propose an adaptation of the Ranking Loss for evaluating the quality of the label example ranking. For consolidating our conclusions, we additionally consider five classical measures: Coverage, One Error, Average Precision, Hamming Loss and Exact match. The computational efficiency of the classifiers is measured by both the running time observed in the experiments and the theoretical computational complexities required for training and testing the models.

The comparison of the twelve algorithms is performed on a set of twelve multi-label benchmarks of various sizes from five different domains (music, audio, image, biology and text). It shows that, for the learning quality criteria, the four classifiers which significantly outperform the others are: the problem transformation method CLR (Calibrated Label Ranking, Fürnkranz et al. (2008)), the algorithm adaptation method ML-kNN (Multi-Label k Nearest Neighbours, Zhang and Zhou (2007)), and the ensemble methods EBR (Ensemble of Binary Relevance, Read et al. (2011)) and RF-PCT (Random Forest of Predictive Clustering Trees, Kocev (2012)). A precise analysis of the difference in their predictive performances concludes that RF-PCT is the best classifier, closely followed by EBR. However, when considering the time criteria, only RF-PCT competes with the fastest classifiers that are the least accurate classifiers. Let us note that RF-PCT was already the best performing classifier for large training data sets (Madjarov et al., 2012).

The rest of the paper is organized as follows. Section 2 presents some related recent works both in interactive classification and in multi-label learning. In Section 3, we precisely define the constraints considered in our interactive multi-label classification problem. The experimental protocol and the benchmark datasets are described in Section 4. Section 5 presents the classifier comparison for the learning efficiency with a limited training set and Section 6 presents the learning and predicting time evaluations.

2. Related Work

We first briefly present some interactive classification systems which have been recently developed. Then, we list the multi-label classifiers used in our comparisons and we recall the performances obtained from published extensive comparisons.

2.1. Interactive classification

In this section, we have retained five recent interactive classification systems to illustrate the potentialities of the user-centered approaches.

***Cue*Flik** (Fogarty et al., 2008) is an image classifier that automatically recognizes a user's desired visual concept (e.g. scenic, visually busy, or colourful images). The user queries a catalogue of images and selects some images with and without the desired visual characteristics from the obtained results. A kNN algorithm re-ranks the images using a similarity measure whose parameters are learned from the user's actions.

Smart selection (Ritter and Basu, 2009) is a file classifier to perform complex file search queries (e.g. selection of all files that contain the substring "old"). To train the classifier, the user only clicks few desired files. Using boosted decision trees, the system selects the rest.

***i*Cluster** (Drucker et al., 2011; Basu et al., 2010) is a document classifier that detects preferred documents. The user defines a set of desired labels and associates each document with the label that better describes it. From a restricted training set, the system provides two predictions: for a new example, a ranking of its predicted labels, and for a selected label, a ranking of its $top(20)$ predicted new examples. It uses a hybrid learning mechanism that combines a logistic regression classifier with a metric learner.

***Cue*T** (Amershi et al., 2011) is an alarm classifier that helps network operators to triage alarms. A operator first defines a set of labels and then manually labels a restricted number of alarms according to their severity. From the triaging decisions, the system uses a nearest

neighbour strategy combined with an adaptive distance function to predict the label of a new incoming alarm in the network.

Regroup (Amershi et al., 2012) is a profile classifier that recognizes desired profiles in a social network. To explain a target profile, a user selects some friends with the desired characteristics. Using a Naïve Bayes classifier, the system re-ranks the remaining friend set according to their probability to belong to the group.

These systems show that interactive classification presents attractive properties for real-life applications. However, all of them constrain users to associate examples with a single label, which seems artificial for decisions on data that generally imply a combination of labels. For instance, images in *Cue*Flik may contain many visual concepts, documents in *i*Cluster may talk about several subjects and users in *Regroup* may belong to different social groups.

2.2. Multi-label classification

Generally speaking, the multi-label classification approaches can be categorized in three main families. The ***problem transformation*** methods are probably the most popular approaches. They do not learn directly from the multi-label data: they transform the multi-label learning problem into one or several single-label classification or regression problems. The ***algorithm adaptation*** methods adapt existing learning algorithms to learn from multi-label data. The ***ensemble*** methods or meta-methods involve a collection of learners to make multi-label predictions. These learners belong to one of the two previous families.

An exhaustive description is beyond the scope of this paper. We here restrict ourselves to a brief presentation of the twelve approaches retained for our comparison. They are among the mostly-studied multi-label classifiers and they include the classifiers that Madjarov et al. (2012) have recommended from their recent extensive experimental study.

2.2.1. Problem transformation methods

We have selected five approaches plus a baseline.

Binary Relevance (BR) (Schapire and Singer, 2000) is probably the most popular transformation method. It learns one binary classifier for each label independently. For a new instance, it outputs the union of the labels predicted by the learned models.

Classifier Chain (CC) (Read et al., 2011) is an extension of BR that trains the classifiers in a random chain and extends the feature space associated with each classifier with the labels of the previous classifiers in the chain. For a new instance, like BR, CC presents the union of the labels predicted by each classifier in the chain.

Label Powerset (LP) (Tsoumakas and Katakis, 2007) considers each label set as a single atomic label and then trains a single-label multi-class classifier. For a new instance, LP predicts the most likely label set.

Calibrated Label Ranking (CLR) (Fürnkranz et al., 2008) extends the Ranking by Pairwise Comparison method (***RPC***) (Hüllermeier et al., 2008) by introducing an additional virtual label to separate the relevant labels from the irrelevant ones. For a new instance, CLR returns the average vote on all models for each label.

Hierarchy Of Multi-label classifiERs (HOMER) (Tsoumakas et al., 2008) recursively constructs a tree of LP classifiers which consider small label subsets. The labels of each node are distributed into several disjoint subsets using a balanced clustering algorithm such that each child node is associated with a different cluster. For a new instance, HOMER starts with the root classifier and follows a recursive process forwarding this instance to the multi-label classifiers of the child nodes.

The ***Baseline*** computes the frequency of each label set in the training set. For a new instance, it returns the most frequent label set.

The problem transformation methods are flexible: they are free to use any existing single-label base classifier. However, their efficiency mainly depends on the choice of this

base learner. In general, two base-learners are commonly used (Tawiah and Sheng, 2013; Madjarov et al., 2012; Read, 2010): Support Vector Machine (SVM) (Platt, 1999) and C4.5 decision tree (Quinlan, 1993) with an advantage to SVM in terms of prediction quality. This choice has been further confirmed in an extensive comparative study of the effect of single-label classifiers on problem transformation methods (Read, 2010). However, in this study, the computation time constraint was not critical (up to 24 hours). Here, due to the strong computation time constraint in our interactive learning framework, we have selected C4.5 decision tree for its lower computational complexity: unlike SVM, it only requires a selected number of features to build a predictive model.

2.2.2. Algorithm adaptation methods

In this second group, we select two adaptation methods.

Multi-Label k Nearest Neighbours (ML-kNN) (Zhang and Zhou, 2007) is a binary relevance method which combines the standard lazy learning algorithm kNN and the Bayesian inference. As a lazy learner, ML-kNN does not learn a model but only estimates the prior and posterior probabilities from the training data. For a new instance, it retrieves its k nearest examples and measures the frequency of each label in this neighborhood. It combines this frequency with the estimated probabilities and finally determines its label set from the maximum a posteriori principle.

Instance-Based learning as Logistic Regression for the Multi-Label case (IBLR_ML) (Cheng and Hüllermeier, 2009) is an extension of MLkNN that combines instance-based learning and logistic regression. Unlike MLkNN, it allows to capture the potentially existing interdependencies between the labels: it uses the labels of neighbours as additional features in a meta logistic regression scheme. For a new instance, as a binary relevance approach, it combines the predictions of all learned models.

2.2.3. Ensemble methods

In this third group, we select four ensemble methods.

RAndom k-labELsets (RAkEL) (Tsoumakas and Vlahavas, 2007) generates an ensemble of LP classifiers. Each LP classifier is trained with a different random label subset of small size. For a new instance, RAkEL outputs the average vote on all models for each label.

Ensemble of Classifier Chains (ECC) and ***Ensemble of Binary Relevance (EBR)*** (Read et al., 2011) are ensemble methods that use a bagging strategy with CC and BR respectively. For a new instance, ECC and EBR output the average votes on all models for each label, like RAkEL.

Random Forest of Predictive Clustering Trees (RF-PCT) (Kocev et al., 2007; Kocev, 2012) is an ensemble method that uses Predictive Clustering Trees (PCTs) as base classifiers. PCTs use a standard top-down induction of decision trees. For diversity, classifiers are trained with a bagging strategy and by selecting a random subset of the feature set at each node of the trees. For a new instance, the predictions of all decision trees are summed using a distribution vote approach.

2.3. Comparative studies of multi-label learning algorithms

Experimental studies have evaluated the predictive performances of the different algorithms in different contexts. Table 1 recalls the results of the main publications where each study (*Reference*) is described by its number of classifiers (*#Classifiers*), number of datasets (*#Datasets*), number of evaluation criteria (*#Criteria*) and the set of recommended classifiers (*Recommendation*). Let us note that, in the quoted studies, the classifiers are always trained on a large number of examples, and that the computation time limitation is very loose (up to many days in some cases). The recent study of Madjarov et al. (2012) which stands out of the others for its comprehensiveness and extensiveness is our point of reference.

Reference	#Classifiers	#Criteria	#Datasets	Recommendation
(Li et al., 2006)	6	9	2	BR and ML_ADTree[1]
(Nasierding and Kouzani, 2012)	7	4	8	TREMLC[2], MLkNN and BR
(Madjarov et al., 2012)	12	16	11	RF-PCT, HOMER, BR and CC
(Tawiah and Sheng, 2013)	6	5	11	MLkNN, RAkEL, CC and BR

Table 1: The most relevant experimental studies of multi-label learning algorithms.

3. Problem statement

In this section, we first formally define the considered learning problem. Then, we precise the selected performance measures for the two main interactivity constraints: learning to generalize from a limited training set and learning and predicting in a very limited time.

3.1. Learning model

In the following, we consider a set $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$ of m numerical features f_i ($dom(f_i) \in \mathcal{R}$), and a set $\mathcal{D} = \{x_1, x_2, \ldots, x_n\}$ of n unlabelled examples x_i described by the m features ($dom(x_j) \in \mathcal{R}^m$). At the beginning t_0 of the process, we assume that the user defines a set $\mathcal{L}_{t_0} = \{\lambda_1, \lambda_2, \ldots, \lambda_q\}$ of q desired labels λ_i ($dom(\lambda_i) \in \{0, 1\}$) and that he selects a small set $\mathcal{T}_{t_0}$ of n_0 examples that he/she labels either positively or negatively. More precisely, let y_i ($dom(y_i) \in \{0, 1\}^q$) be the binary vector which describes the labels given to an example x_i: $y_i^j = 1$ (resp. 0) if the label λ_j is positively (resp. negatively) associated to x_i. The set $\mathcal{T}_{t_0}$ of the labelled examples can be defined by $\mathcal{T}_{t_0} = \{(x_i, y_i) |\ i = 1..n_0$ and $|y_i^+| + |y_i^-| = q\}$ where $|y_i^+|$ and $|y_i^-|$ are respectively the number of positive and negative labels of x_i. Gradual relevance can be used to label the examples (e.g. (Cheng et al., 2010)) but we here restrict ourselves to the binary relevance case where the user provides a simple '*yes*' (1) or '*no*' (0) answer.

From the multi-label training set $\mathcal{T}_{t_0}$, a multi-label learning algorithm learns a predictive model h_{t_0}. The learned model predicts the most likely label set $\hat{y}_i = h_{t_0}(x_i)$ for each selected

example $x_i \in \mathcal{S} \subset \mathcal{D}$ where $\mathcal{S}$ is a test set with $|\mathcal{S}| >> |\mathcal{T}_{t_0}|$. If the predictions provided by the model do not align well with the user's preferences, he/she can boost the predictive performance by correcting the mistakes or adding few more examples and the learning process is run again. In our proposed experimental protocol (see subsection 4.1), we do not simulate the user selection and correction. We restrict ourselves to the evaluation of the classifier predictive and time-computation performances while new examples are provided.

Generally, the multi-label learning algorithms predict a vector of real-valued confidence outputs. To transform these real values into binary ones, a threshold function is needed. We here use a fast and effective threshold method *Pcut* (Proportional Cut Method) introduced in (Read, 2010)). It chooses the z value which minimizes the label cardinality difference between the training data set $\mathcal{T}$ and the classified test data set $\mathcal{S}$ where $f_z : [0..1]^q \rightarrow \{0,1\}^q$ is a threshold function that turns values greater than z into ones (1) or zeros (0) otherwise:

$$PCut = \underset{z \in \{0.00,0.001,...,1.00\}}{\operatorname{argmin}} \left| \frac{1}{|\mathcal{T}|} \sum_{i=0}^{|\mathcal{T}|} |y_i^+| - \frac{1}{|\mathcal{S}|} \sum_{i=0}^{|\mathcal{S}|} |f_z(\hat{y}_i)^+| \right|$$

This threshold method was mainly used for large training sets. However, we here assume that the average number of labels in a limited or a large training set is not significantly different.

3.2. Constraint 1 : Learning to generalize from a limited training set

From decision theory, it is well-known that users have a limited focus when it comes to making decisions (Simon, 1955). Asking a user to provide a large number of examples to explain a desired concept is consequently a hard task which must be avoided. For the evaluation of the classifier efficiency with a limited training set, we consider five complementary measures relevant for the requirements that are considered in priority in multi-label learning applications: ranking labels by relevance, ranking examples by relevance and classifying labels. Moreover, we consolidate the obtained results with five additional measures classically

used in the literature. Table 2 provides a summary of these evaluation criteria called *quality criteria* in the following. We describe them below.

3.2.1. Requirement 1: Ranking labels by relevance

In practice, users are mostly interested in a label ranking for a given example. Consequently, when an example is selected, the learning system must present its most likely labels at the top of the prediction list. To evaluate the classifier performances for ranking the example labels, we select a common criterion: *Ranking Loss* (*RL*). It is defined in the interval [0..1] and its lowest values indicate the best performances. *RL* measures the number of times where relevant and irrelevant labels are reversely ordered. Formally, let r_i be a ranking function that sorts the labels of each example x_i in descending order with respect to their prediction precision $\hat{y}_i : r_i(\lambda_a) = k, k \in \{1, 2, \ldots, q\}$, if $\hat{y}_i^a$ is the k^{th} larger value among the $\hat{y}_i$ values. The *RL* of a classifier on a test set $\mathcal{S}$ is defined by

$$RL = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \frac{1}{|y_i^+| \times |y_i^-|} |(\lambda_a, \lambda_b) \in y_i^+ \times y_i^- : r_i(\lambda_b) < r_i(\lambda_a)|$$

3.2.2. Requirement 2: Ranking examples by relevance

Users can also be interested in an example ranking for one or a set of labels. Therefore, when a label or a combination of labels is selected, the learning system must present its most likely examples at the top of the prediction list. To evaluate the classifier performances for ranking the label examples, we have adapted the RL definition and we define the macro-averaged Ranking-Loss criterion (*macro-RL*) which measures the number of times that relevant and irrelevant examples are reversely ordered. As for the RL definition, macro-RL is defined in the interval [0..1] and its lowest values indicate the best performances. Formally, let $|\gamma_i^+|$ and $|\gamma_i^-|$ be respectively the number of positive and negative examples associated with the label λ_i. Let $\hat{\gamma}_i$ ($dom(\hat{\gamma}_i) \in [0..1]^{|\mathcal{S}|}$) be the real-valued vector which describes the prediction precisions associated with the examples $x_i \in \mathcal{S}$ for the label λ_i. Then, let r'_i be a

ranking function that sorts each vector $\hat{\gamma}_i$ in descending order: $r'_i(x_a) = k, k \in \{1, 2, \ldots, |\mathcal{S}|\}$, if $\hat{\gamma}_i^a$ is the k^{th} larger value among the $\hat{\gamma}_i$ values. The *macro-RL* of a classifier on a test set $\mathcal{S}$ is defined by

$$macro - RL = \frac{1}{|\mathcal{L}|} \sum_{i=1}^{|\mathcal{L}|} \frac{1}{|\gamma_i^+| \times |\gamma_i^-|} |(x_a, x_b) \in \gamma_i^+ \times \gamma_i^- : r'_i(x_b) < r'_i(x_a)|$$

3.2.3. Requirement 3: Label classification

As previously mentioned, a label ranking is essential in a multi-label learning system but a label classification may be sometimes desired. Consequently, when an example is selected, the learning system must only present its most likely labels. To evaluate the classifier performances to correctly classify the example labels, we select three criteria: *Accuracy* and F_1*-score* and the multi-label Balanced Error Rate (*BER*). In Madjarov et al. (2012), the *Accuracy* and the F_1*-score* criteria help to detect the best classifiers and the BER criterion is adapted when the evaluation datasets are unbalanced (e.g. *Slashdot* (Read et al., 2011)).

The ***Accuracy*** (Godbole and Sarawagi, 2004) for a single example x_i is defined by the Jaccard similarity coefficient between its ground truth y_i and the predicted label set $\hat{y}_i$. More precisely, it is defined by

$$Accuracy = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \frac{|y_i \cap \hat{y}_i|}{|y_i \cup \hat{y}_i|}$$

The F_1***-score*** (Spyromitros et al., 2008; Tsoumakas and Katakis, 2007) is commonly used in information retrieval but it is also popular in multi-label classification. It is the harmonic mean between the precision and the recall criteria of each example x_i:

$$F_1 - score = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \frac{2 \times |y_i \cap \hat{y}_i|}{|y_i^+| + |\hat{y}_i^+|}$$

The ***BER*** (Chen and Lin, 2006) has mostly been used to evaluate single-label predictions but we here adapt it for the evaluation of multi-label predictions. It is defined by the ratio

of incorrectly classified labels per example where TP_i, TN_i, FP_i and FN_i are the number of respectively true positive, true negative, false positive and false negative labels of an example x_i:

$$BER = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \frac{1}{2} \times \left(\frac{FP_i}{FP_i + TN_i} + \frac{FN_i}{FN_i + TP_i} \right)$$

All these criteria are defined in the interval [0..1] and their highest values indicate the best performances except for the ***BER*** criterion whose smallest value indicate the best performances.

3.2.4. Additional quality criteria

In a previous work on multi-label classification, Tsoumakas et al. (2010) have organized the quality criteria into two groups: (i) the ranking-based measures which compare the predicted label ranking with the ground truth label ranking and (ii) the bipartition-based measures which are based on the comparison of the predicted relevant labels with the ground truth relevant labels. In order to limit bias induced by quality criteria selection in the conclusion, we have added five well-known quality criteria from each group: (i) Coverage, One-error and Average precision and (ii) Hamming lost and Exact match. Criteria from (i) (resp. (ii)) contribute to the evaluation of the requirement 1 Section 3.2.1 (resp. requirement 3 Section 3.2.3). Their definitions are recalled in Appendix 2.

Quality Criteria	Abbreviation	Min/Max	Domain	Ranking/Classification
Ranking Loss	RL	Min	[0..1]	Ranking-based
Macro-averaged Ranking Loss	macro-RL	Min	[0..1]	Ranking-based
Coverage	Coverage	Min	$[0..\|\mathcal{L}\|]$	Ranking-based
One error	One error	Min	[0..1]	Ranking-based
Average precision	Average precision	Max	[0..1]	Ranking-based
Accuracy	Accuracy	Max	[0..1]	Bipartition-based
Hamming Loss	Hamming Loss	Min	[0..1]	Bipartition-based
Exact match	Exact match	Max	[0..1]	Bipartition-based
F_1-score	F_1-score	Max	[0..1]	Bipartition-based
Balanced Error Rate	BER	Min	[0..1]	Bipartition-based

Table 2: A summary of the selected criteria for the evaluation of the prediction quality.

3.3. Constraint 2 : Learning and predicting in a very limited time

In an interactive framework, the response of the learning system must be short: whenever a user adds new examples, the learning system must quickly adjust its current understanding and provide him/her with predictions as fast as possible. In Human-Computer Interaction, interactive systems are often required to provide users with a response in less than 100 ms (Dabrowski and Munson, 2001); as far as we know, this constraint is currently too strong here and we relax it to few seconds. For each classifier, we compute the number of seconds required to learn from a limited training data, and we predict labelsets of large test data. Obviously, we are aware that the computation time mainly depends on the implementation of the classifiers, and that the obtained results might only provide tendencies of their computational complexities.

4. Experimental setting

We first describe the experimental protocol proposed to evaluate the classifier performances in a simplified interactive context which allows classifier comparisons under the

same conditions. Then, we present the twelve multi-label classification benchmark problems used for the evaluations and the chosen classifier parameters.

4.1. Experimental protocol

Evaluating the relevance of classifiers for an interactive environment is a complex task and, unfortunately, due to the novelty of this research field, there is still no standard or widely accepted framework to compare different approaches. In works presented in Section 2.1, the evaluation is mainly subjective: a small sample of users rated the quality of the developed systems for different tasks. Some authors additionally introduced "objective" measures (e.g. prediction accuracy, average trial time and learning speed). However, they only considered just one or a very restricted set of classification algorithms –often chosen without solid arguments- and few datasets –often one only-. The experimental evaluations closely depend on the chosen targeted use cases and the interface characteristics. The significance of their conclusions is consequently limited.

To draw general conclusions helpful for guiding the choice of a suitable classifier during the development of an interactive multi-label classification system, we use a simplified simulation where the training sets regularly increase while staying small. The objective is to detect the classifiers which are able to "continuously" learn well with very limited training sets in a reasonable time. More precisely, we focus on the beginning of the classification task where few examples (from 2 to 64 examples) are available. In practice, this phase is crucial for catching the user interest and confidence in the system.

To avoid bias in the comparisons, all the classifiers are trained with the same training examples. The principle of the experimental protocol is the following. Each dataset is divided into a small training set and a large test set. From each training set, training subsets of restricted sizes are successively created such that each one fits into the other. Thereafter, each classifier is trained with the nested data subsets and its performances are evaluated for each training data subset size on the same test set. This process allows to

precisely follow the evolution of the classifier performances while the training set grows. It is repeated several times as precisely described below.

1. Divide each dataset $\mathcal{D}$ into 5 distinct folds. Use each fold for training ($\mathcal{T}$ = 20% of $\mathcal{D}$) and the 4 remaining folds for testing ($\mathcal{S}$ = 80% of $\mathcal{D}$). In total, there are 5 sets $\mathcal{T}_i$ (i = 1 to 5) for training and 5 sets $\mathcal{S}_i$ (i = 1 to 5) for testing (i.e. 5 cross-validation).
2. From each training set $\mathcal{T}_{i\ (1\leq i\leq 5)}$, extract s sets of p nested training subsets of size 2^1, 2^2, ... to 2^p.
3. Associate each classifier with the $5 \times s \times p$ training subsets of all folds ($5 \times s$ training data sets for each size). For each training subset size and for each criterion, evaluate its average performance for the 5 test sets. Then, average on all datasets.

For all the experiments, s was fixed to 10 and p to 6, which corresponds to a number of 300 train-test evaluations for the 5-cross validation. The threshold ($p = 6$) is consistent with real-life experiments. From our practical experience, we have observed that users do not annotate more than 64 examples by themselves without any assistance of a learning algorithm.

This online learning approach is adapted for the beginning of the interaction where the model is more likely to change (i.e. concept-drift). In practice, users define their desired concepts mainly in real time while interacting with data and the learning system, and they mostly have no clear idea about the concepts they have in mind before the interaction starts (i.e. concept flexibility (Amershi, 2011)).

4.2. Data sets

We use twelve different multi-label classification benchmark problems most of which were selected in various previous studies. Our experimental corpus includes datasets with different scale from five different application domains (music, image, audio, biology, text). The basic statistics (Table 3) confirm that they cover a wide range of situations. In particular, their

number of features varies from 71 to 49060 which allows to evaluate the classifier behaviours in a large scale, and the number of labels varies from 6 to 30 labels as users are mostly interested in a limited number of labels. The datasets are briefly described below.

Emotions (Trohidis et al., 2008) is a small dataset which describes pieces of music by 71 numerical features. They can be labelled with 6 possible emotions: sad-lonely, angry-aggressive, amazed-surprised, relaxing-calm, quiet-still, and happy-pleased. ***Yeast*** (Elisseeff and Weston, 2001) is a biological dataset where genes are described by 103 numerical features. They can be associated with 14 biological functions. ***Scene*** (Boutell et al., 2004) is a dataset where images are described by 294 numerical features. They can be annotated with up to 6 concepts: beach, sunset, field, fall-foliage, mountain, and urban. *Birds* (Briggs et al., 2013) is a small dataset where 645 ten-second audio recordings of bird sounds are described by 260 numerical features. They can be labelled with up to 19 bird species. ***Slashdot*** (Read et al., 2011) is a sparse text dataset where documents are defined by 1079 binary features. They can be associated with 20 subject categories (e.g. linux, technology, science). ***IMDB*** (Read et al., 2011) is a sparse dataset where movies are defined by 1001 binary features. They can be tagged with up to 28 genres (e.g. Romance, Comedy, Drama). *Genbase* (Diplaris et al., 2005) is another microbiological dataset where genes are described with 1186 binary features. They can be associated with 27 biological functions.

TMC (Srivastava and Zane-Ulman, 2005) is a sparse text dataset of flight readiness and discrepancy reports. It is described by 49060 binary features. The reports can be associated with up to 22 labels representing the problems being described. The four remaining text datasets (***Arts***, ***Business***, ***Health*** and ***Computers***) are web pages collected through the hyperlinks from Yahoo!'s top directory. Each data set is associated with four of Yahoo!'s top categories ("Arts & Humanities", "Business & Economy", "Computers & Internet", "Health"), and each page is labelled with one or more second level subcategories. In these four datasets, the minimum (maximum) values of $|\mathcal{L}|$ and $|\mathcal{F}|$ are 24 (30) and 21924 (34096), respectively.

The diversity of the data sets considered in our experimentation is confirmed by the distributions of three classical criteria which measure their « multi-labelled-ness » (see Table 3 for the numerical values):

1. ***Label Cardinality (LCard)*** is arguably the most common measure of multi-labelled-ness in the literature (Tsoumakas and Katakis, 2007). It evaluates the average number of labels associated with each example in a dataset $\mathcal{D}$:

$$LCard(\mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n}|y_i^+|$$

2. ***Label Density (LDens)*** relates to *LCard*, but takes into account the size of the label space. It is equal to the ratio of the average number of the example labels in a dataset $\mathcal{D}$ by the label number q (Tsoumakas and Katakis, 2007):

$$LDens(\mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i^+|}{q} = \frac{LC}{q}$$

3. ***Proportion of Unique label combinations (PUniq)*** is a new measure of multi-labelled-ness which was recently introduced by (Read, 2010). It indicates the regularity or uniformity of the labelling scheme. Precisely, it evaluates the proportion of label sets which are unique across the total number of examples in a dataset $\mathcal{D}$:

$$PUniq(D) = \frac{|y_i|\exists!\ x_i : (x_i, y_i) \in \mathcal{D}|}{n}$$

In our selected evaluation datasets, *LCard* values are mostly smaller than 2.0 except for Yeast where examples are associated in average with more than 4.0 labels. Indeed, the low label cardinality is common to textual and multi-media data where most examples are associated with a single-label, and the "multi-labelled-ness" has only been used to avoid ambiguities. *LDens* values are mostly very low because labelling is usually very sparse except for Emotions and Yeast where 30% of the labels are associated on average with each

example. The low values of *PUniq* indicate that the labelling is generally regular except for IMDB and TMC where more than 20% of the examples are associated with unique labelsets (i.e. irregular labelling).

In an interactive classification framework, waiting for the successive results must be reduced. Therefore, to obtain real estimations of the average learning and prediction times of the classifier, we here only select random samples of 1000 examples from the original datasets -expect for Emotions which is already small.

Dataset	Domain	$\|\mathcal{F}\|$	$\|\mathcal{D}\|$	$\|\mathcal{L}\|$	*LCard*	*PUniq*	*LDens*
Emotions	Music	71	592	6	1.86	0.04	0.31
Yeast	Biology	103	1000	14	4.2	0.14	0.30
Scene	Image	294	1000	6	1.07	0.01	0.18
Birds	Audio	260	645	19	1.01	0.21	0.05
Slashdot	Text	1079	1000	20	1.19	0.09	0.06
IMDB	Text	1001	1000	28	1.94	0.27	0.07
Genbase	Biology	1186	662	27	1.25	0.05	0.05
Arts	Text	23146	1000	24	1.66	0.18	0.07
Business	Text	21924	1000	28	1.55	0.07	0.05
Health	Text	30605	1000	25	1.63	0.11	0.06
Computers	Text	34096	1000	30	1.44	0.10	0.05
TMC	Text	49060	1000	22	2.18	0.23	0.1

Table 3: Basic statistics of the selected multi-label benchmarks ($|\mathcal{F}|$: number of features, $|\mathcal{D}|$: number of examples, $|\mathcal{L}|$: number of labels, *LCard*: Label Cardinality, *PUniq*: Proportion of Unique label combinations, *LDens*: Label Density)

4.3. Classifier parameters

The selected classifiers have been described in section 2.2. The parameters chosen for the experiments follow the recommendations from the literature -except for three approaches: for HOMER which was trained with the implementation's default parameters, and for MLkNN and IBLR_ML, we set the number of neighbours to 1 as the number of training examples is very limited. The classifier parameters are precised in Table 4. Implementations of the selected classifiers are available in the following multi-label machine learning libraries that

are most widely used in the literature: *MeKA*[3], *MULAN*[4] and *CLUS*[5].

Classifier	parameters	Library	Reference
LP	/	MeKA	/
CC	/	MeKA	/
ECC	$N = 10 \times q$	MeKA	Read et al., 2009
BR	/	MeKA	/
EBR	$N = 10 \times q$	MeKA	Read et al., 2009
RAkEL$_1$	$N = 10$ and $k = q/2$	MULAN	Tsoumakas et al., 2007
RAkEL$_2$	$N = 2 \times q$ and $k = 3$	MULAN	Tsoumakas et al., 2007
MLkNN	$k = 1$	MULAN	ours
IBLR_ML	$k = 1$	MULAN	ours
HOMER	$k = 3$	MULAN	default
CLR	/	MULAN	/
RF-PCT	$N = 100$, $m' = 0.1 \times \|\mathcal{F}\| + 1$	CLUS	Kocev 2011

Table 4: The input parameters of each multi-label classifier where q is the number of labels, N is the number of base learners for the ensemble methods, m' is the number of features selected at each node in RF-PCT, and k could be the number of label subsets, the size of label subsets or the number of neighbours respectively for HOMER, RAkEL and instance-based methods.

5. Experimental results I: learning from a limited training set

Tables 5-9 present the results obtained for the quality criteria defined in subsection 3.2. The average performance of each classifier is given for each training data size (from 2 to 64). The classifier predictive performances significantly improve as new training examples are provided, especially when the training data size is greater than 8 -except for the macro-RL. Let us note that, unsurprisingly, they all struggle to learn from the smallest training sets of size 2 –which explains their very close poor performances-. Moreover, the differences between the classifier performances increase with the number of training examples. The differences are confirmed by a Friedman statistical test (with a 5% significance level) for all criteria but the BER criterion.

[3]meka.sourceforge.net
[4]mulan.sourceforge.net
[5]dtai.cs.kuleuven.be/clus

For the major quality criteria (RL and macro-RL), the ensemble method RF-PCT outperforms the other classifiers for all training data sizes (Tables 5-6). It is closely followed by EBR, CLR and MLkNN which belong to the three multi-label method families. The Nemenyi post-hoc analysis does not reveal any statistical difference between these classifiers. The detailed results obtained for three representative training data sizes (4, 16 and 64 examples) show that the best classifiers remain the same whatever the dataset (see Appendix 1 – Tables .13 to .18); this is confirmed by a Friedman statistical test.

When considering the other quality criteria (Accuracy, F_1-score and BER), the podium remains the same and RF-PCT remains the winner (Tables 7 and 9). Let us recall that RF-PCT and CLR were already among the best multi-label classifiers for the RL, Accuracy and F_1-score criteria in the extensive study of Madjarov et al. (2012) which does not take the interactive constraints into account. In the previous study, EBR was not evaluated and MLkNN only performed well for the RL criterion with poor results for the Accuracy and the F_1-score. The critical diagrams obtained from the statistical tests (Friedman and Nemenyi post-hoc) for the main quality criteria are given in Appendix 3 (Figures .2-.4).

The main conclusion is that the learning capabilities of the ensemble methods RF-PCT and EBR remain good whatever the training set size. Additional details are given bellow for each criterion.

5.1. Ranking Loss (RL)

A precise analysis places RF-PCT first for all training data sizes. It is followed by CLR, EBR and MLkNN which obtain very close performances with a very slight advantage to CLR intrinsically optimizes this criterion; this is confirmed by a Friedman statistical test. In contrast, BR and CC, which were favourite in Madjarov et al. (2012), loose their effectiveness for small training sets. Moreover, when the number of training examples increase, each classifier was able to improve its ranking performance - except HOMER which has also previously provided poor performances for large training data sets. It seems that HOMER

is more adapted for domains with a large label number (hundreds and more) (Tsoumakas et al., 2008).

Some remarks can also be drawn for the different classifier families. For the ensemble learning methods, EBR is better than a single BR, ECC is slightly better than a single CC and RAkEL1 slightly outperforms RAkEL$_2$. For the problem transformation methods, CC does not outperform BR and they obtain very close performances as in Madjarov et al. (2012). For the algorithm adaptation methods, MLkNN outperforms IBLR_ML and this result differs from previous results obtained by (Cheng and Hüllermeier, 2009) with large training data sets.

	2	4	8	16	32	64
Baseline	$0,39 \pm 0,13$	$0,37 \pm 0,14$	$0,35 \pm 0,14$	$0,34 \pm 0,13$	$0,34 \pm 0,13$	$0,33 \pm 0,13$
LP	**0,34 ± 0,13**	$0,34 \pm 0,11$	$0,32 \pm 0,11$	$0,30 \pm 0,11$	$0,28 \pm 0,11$	$0,26 \pm 0,11$
CC	**0,34 ± 0,13**	$0,33 \pm 0,12$	$0,31 \pm 0,11$	$0,29 \pm 0,10$	$0,26 \pm 0,11$	$0,24 \pm 0,11$
RAkEL1	$0,35 \pm 0,13$	$0,34 \pm 0,12$	$0,32 \pm 0,12$	$0,29 \pm 0,11$	$0,26 \pm 0,11$	$0,24 \pm 0,11$
RAkEL2	$0,35 \pm 0,12$	$0,36 \pm 0,13$	$0,34 \pm 0,13$	$0,31 \pm 0,13$	$0,28 \pm 0,13$	$0,25 \pm 0,13$
MLkNN	**0,34 ± 0,13**	$0,32 \pm 0,13$	$0,28 \pm 0,11$	$0,24 \pm 0,09$	$0,20 \pm 0,08$	$0,18 \pm 0,08$
HOMER	$0,43 \pm 0,10$	$0,41 \pm 0,11$	$0,39 \pm 0,12$	$0,38 \pm 0,13$	$0,37 \pm 0,14$	$0,35 \pm 0,15$
IBLR-ML	**0,34 ± 0,13**	$0,32 \pm 0,13$	$0,30 \pm 0,11$	$0,26 \pm 0,10$	$0,23 \pm 0,09$	$0,20 \pm 0,08$
CLR	**0,34 ± 0,13**	**0,31 ± 0,13**	$0,28 \pm 0,12$	$0,24 \pm 0,10$	$0,20 \pm 0,07$	$0,17 \pm 0,07$
ECC	$0,37 \pm 0,13$	$0,33 \pm 0,13$	$0,31 \pm 0,13$	$0,28 \pm 0,13$	$0,25 \pm 0,12$	$0,22 \pm 0,11$
BR	**0,34 ± 0,13**	$0,33 \pm 0,12$	$0,31 \pm 0,11$	$0,28 \pm 0,10$	$0,25 \pm 0,10$	$0,23 \pm 0,10$
EBR	**0,34 ± 0,13**	$0,32 \pm 0,12$	$0,28 \pm 0,11$	$0,24 \pm 0,09$	$0,20 \pm 0,07$	$0,17 \pm 0,07$
RF-PCT	**0,34 ± 0,13**	**0,31 ± 0,12**	**0,27 ± 0,11**	**0,23 ± 0,08**	**0,18 ± 0,06**	**0,15 ± 0,06**

Table 5: The average performances of each classifier for each training set size for the Ranking Loss criterion (RL).

5.2. Macro-averaged Ranking Loss (macro-RL)

To our knowledge, the ability of multi-label classifiers to correctly rank examples of each label has not yet been evaluated. Table 6 shows that the ensemble method RF-PCT is clearly the best for all training set sizes -except for the smallest size where all classifiers suffer. It is followed by the ensemble methods ECC and EBR. This is confirmed by a

Friedman statistical test. However, when the size of the training set increases, all classifiers struggle to improve their predictions except RF-PCT. It seems that the multi-label learning algorithms inherently focus on the label ranking and do not consider the example ranking.

	2	4	8	16	32	64
Baseline	**0,49 ± 0,02**	0,49 ± 0,02	0,49 ± 0,02	0,49 ± 0,02	0,49 ± 0,02	0,49 ± 0,02
LP	**0,49 ± 0,02**	0,48 ± 0,02	0,47 ± 0,03	0,45 ± 0,04	0,43 ± 0,07	0,41 ± 0,08
CC	**0,49 ± 0,02**	0,48 ± 0,02	0,47 ± 0,02	0,46 ± 0,03	0,44 ± 0,05	0,43 ± 0,07
RAkEL1	**0,49 ± 0,02**	0,48 ± 0,02	0,47 ± 0,03	0,44 ± 0,05	0,42 ± 0,07	0,39 ± 0,09
RAkEL2	**0,49 ± 0,02**	0,48 ± 0,02	0,47 ± 0,03	0,44 ± 0,05	0,42 ± 0,08	0,39 ± 0,09
MLkNN	**0,49 ± 0,02**	0,49 ± 0,02	0,47 ± 0,03	0,45 ± 0,06	0,43 ± 0,08	0,42 ± 0,10
HOMER	**0,49 ± 0,02**	0,48 ± 0,02	0,47 ± 0,03	0,45 ± 0,05	0,44 ± 0,07	0,43 ± 0,08
IBLR-ML	**0,49 ± 0,02**	0,49 ± 0,02	0,48 ± 0,02	0,47 ± 0,04	0,44 ± 0,07	0,42 ± 0,11
CLR	**0,49 ± 0,02**	0,48 ± 0,02	0,47 ± 0,03	0,45 ± 0,05	0,42 ± 0,08	0,38 ± 0,10
ECC	**0,49 ± 0,02**	0,48 ± 0,02	0,46 ± 0,04	0,43 ± 0,07	0,40 ± 0,09	0,38 ± 0,10
BR	**0,49 ± 0,02**	0,48 ± 0,02	0,47 ± 0,02	0,46 ± 0,04	0,44 ± 0,06	0,42 ± 0,07
EBR	**0,49 ± 0,02**	0,48 ± 0,02	0,46 ± 0,04	0,43 ± 0,07	0,40 ± 0,09	0,38 ± 0,10
RF-PCT	**0,49 ± 0,02**	**0,46 ± 0,03**	**0,43 ± 0,06**	**0,39 ± 0,09**	**0,35 ± 0,11**	**0,32 ± 0,12**

Table 6: The average performances of each classifier for each training set size for the macro-averaged Ranking Loss criterion (macro-RL).

5.3. Accuracy, F_1-score and BER

As the ranking measures have priority in an interactive multi-label classification framework, we only retain the best classifiers according to RL and macro-RL to deepen our understanding of their behaviours with the other evaluation criteria (accuracy, F_1-score, BER). For the accuracy and the F_1-score (Tables 7 and 8), RF-PCT is still the most efficient for all training set sizes except for the smallest training set size where CLR is more efficient. Nemenyi post-hoc tests confirm that RF-PCT is significantly better than MLkNN for the F_1-score and the Accuracy. However, for the BER criterion (Table 9), the classifiers obtain very close performances for all training data sizes which is confirmed by a Friedman statistical test.

	2	4	8	16	32	64
MLkNN	$0,17 \pm 0,10$	$0,20 \pm 0,10$	$0,24 \pm 0,11$	$0,28 \pm 0,13$	$0,31 \pm 0,13$	$0,33 \pm 0,16$
CLR	$\mathbf{0,19 \pm 0,10}$	$0,21 \pm 0,11$	$0,24 \pm 0,12$	$0,26 \pm 0,12$	$0,30 \pm 0,12$	$0,34 \pm 0,13$
EBR	$0,18 \pm 0,10$	$0,20 \pm 0,10$	$0,26 \pm 0,11$	$0,30 \pm 0,13$	$0,33 \pm 0,14$	$0,38 \pm 0,16$
RF-PCT	$0,18 \pm 0,10$	$\mathbf{0,24 \pm 0,12}$	$\mathbf{0,28 \pm 0,13}$	$\mathbf{0,33 \pm 0,12}$	$\mathbf{0,39 \pm 0,13}$	$\mathbf{0,44 \pm 0,17}$

Table 7: The average performances of the $top(4)$ classifiers for each training set size for the Accuracy criterion.

	2	4	8	16	32	64
MLkNN	$0,22 \pm 0,14$	$0,27 \pm 0,13$	$0,32 \pm 0,14$	$0,36 \pm 0,15$	$0,40 \pm 0,16$	$0,42 \pm 0,18$
CLR	$\mathbf{0,26 \pm 0,14}$	$0,28 \pm 0,15$	$0,31 \pm 0,15$	$0,35 \pm 0,15$	$0,40 \pm 0,15$	$0,43 \pm 0,15$
EBR	$0,24 \pm 0,13$	$0,27 \pm 0,13$	$0,33 \pm 0,14$	$0,37 \pm 0,15$	$0,40 \pm 0,16$	$0,45 \pm 0,18$
RF-PCT	$0,24 \pm 0,14$	$\mathbf{0,29 \pm 0,14}$	$\mathbf{0,34 \pm 0,15}$	$\mathbf{0,39 \pm 0,14}$	$\mathbf{0,45 \pm 0,15}$	$\mathbf{0,51 \pm 0,17}$

Table 8: The average performances of the $top(4)$ classifiers for each training set size for the F_1-score criterion.

	2	4	8	16	32	64
MLkNN	$0,40 \pm 0,10$	$0,38 \pm 0,10$	$\mathbf{0,34 \pm 0,10}$	$\mathbf{0,31 \pm 0,10}$	$0,28 \pm 0,10$	$0,27 \pm 0,11$
CLR	$\mathbf{0,38 \pm 0,11}$	$\mathbf{0,37 \pm 0,10}$	$\mathbf{0,34 \pm 0,10}$	$0,32 \pm 0,09$	$\mathbf{0,28 \pm 0,09}$	$0,26 \pm 0,09$
EBR	$0,39 \pm 0,10$	$\mathbf{0,37 \pm 0,09}$	$\mathbf{0,34 \pm 0,10}$	$0,32 \pm 0,10$	$0,30 \pm 0,10$	$0,28 \pm 0,10$
RF-PCT	$0,39 \pm 0,10$	$\mathbf{0,37 \pm 0,09}$	$0,35 \pm 0,09$	$0,32 \pm 0,08$	$\mathbf{0,28 \pm 0,08}$	$\mathbf{0,25 \pm 0,09}$

Table 9: The average predictive performance of the $top(4)$ classifiers for each training set size for the BER criterion.

Let us note that our conclusions are consolidated by the results obtained for the additional quality criteria introduced in Section 3.2.4 (see Appendix 2). The best results are still overall obtained by the ensemble methods RF-PCT and EBR (see Tables .19-.23).

6. Experimental results II: Learning and Predicting in a limited time

Tables 10 and 11 present the average number of seconds measured for testing and training the classifiers. For the prediction time, four classifiers stand out of the rest of classifiers: two ensemble methods RF-PCT and RAkEL$_1$ and two problem transformation methods LP

and HOMER. For the training time, three problem transformation methods LP, CC and BR obtain good performances. They are followed by the ensemble method RF-PCT. If the observed training times are globally consistent with the theoretical computational complexities (Table 12), the observed prediction times are more surprising. In particular, RF-PCT, which requires hundred decision trees for the prediction, was expected to be less efficient than LP, HOMER and RAkEL$_1$ which require a smaller set of decision trees. Moreover, the positive correlation between the prediction time increasing with the number of training examples is only checked for the adaptation methods MLkNN and IBLR_ML.

These amazing experimental results may be partly explained by the coding quality heterogeneity of the algorithms which come from three different libraries (MULAN, MeKA, CLUS). Here, we have followed the previous comparisons in the literature by using the well-known libraries of the community. A coding standardization will be necessary in the future for further analysis but this discussion opens questions that go far beyond the objective of this paper. Nevertheless, combining the theoretical complexity and the experimental results, we suggest to retain the problem transformation methods LP, BR and CC and the ensemble method RF-PCT which is efficient with the code provided in CLUS.

	2	4	8	16	32	64
Baseline	$0,55 \pm 0,68$	$0,56 \pm 0,69$	$0,54 \pm 0,67$	$0,53 \pm 0,67$	$0,54 \pm 0,67$	$0,53 \pm 0,66$
LP	$1,23 \pm 1,54$	$1,23 \pm 1,54$	$1,24 \pm 1,55$	$1,24 \pm 1,55$	$1,23 \pm 1,54$	$1,24 \pm 1,55$
CC	$8,90 \pm 11,3$	$8,92 \pm 11,4$	$8,91 \pm 11,4$	$8,81 \pm 11,2$	$8,80 \pm 11,2$	$8,83 \pm 11,3$
RAkEL1	$2,85 \pm 3,56$	$2,93 \pm 3,67$	$2,94 \pm 3,68$	$2,98 \pm 3,74$	$2,99 \pm 3,74$	$3,01 \pm 3,73$
RAkEL2	$5,27 \pm 6,62$	$5,96 \pm 7,55$	$6,59 \pm 8,26$	$7,26 \pm 9,10$	$7,66 \pm 9,58$	$7,91 \pm 9,90$
MLkNN	$2,66 \pm 3,34$	$3,01 \pm 3,83$	$3,66 \pm 4,82$	$4,67 \pm 6,67$	$6,67 \pm 10$	$10,54 \pm 17$
HOMER	$2,39 \pm 3$	$2,48 \pm 3,12$	$2,54 \pm 3,21$	$2,57 \pm 3,24$	$2,69 \pm 3,42$	$2,65 \pm 3,36$
IBLR-ML	$2,73 \pm 3,43$	$3,10 \pm 3,94$	$3,66 \pm 4,79$	$4,79 \pm 6,70$	$6,71 \pm 10,2$	$10,80 \pm 17,4$
CLR	$21,11 \pm 30,3$	$20,88 \pm 30,2$	$20,60 \pm 29,6$	$21,11 \pm 30,2$	$21,00 \pm 30$	$20,73 \pm 29,5$
ECC	$89,58 \pm 114$	$88,72 \pm 113$	$88,96 \pm 114$	$89,05 \pm 113$	$88,27 \pm 113$	$89,12 \pm 113$
BR	$17,29 \pm 22$	$17,21 \pm 22$	$17,32 \pm 22,1$	$17,08 \pm 21,8$	$17,17 \pm 21,9$	$17,26 \pm 22,1$
EBR	$166,92 \pm 213$	$167,77 \pm 214$	$167,31 \pm 213$	$166,86 \pm 213$	$167,02 \pm 213$	$167,66 \pm 214$
RF-PCT	$\mathbf{0,36 \pm 0,39}$	$\mathbf{0,34 \pm 0,39}$	$\mathbf{0,36 \pm 0,42}$	$\mathbf{0,41 \pm 0,49}$	$\mathbf{0,49 \pm 0,62}$	$\mathbf{0,66 \pm 0,88}$

Table 10: The average prediction times of each classifier for each training set size (in seconds).

	2	4	8	16	32	64
Baseline	$0,00 \pm 0,00$	$0,00 \pm 0,00$	$0,00 \pm 0,00$	$0,00 \pm 0,00$	$0,00 \pm 0,00$	$0,00 \pm 0,00$
LP	$\mathbf{0,04 \pm 0,05}$	$\mathbf{0,05 \pm 0,06}$	$\mathbf{0,07 \pm 0,09}$	$\mathbf{0,13 \pm 0,17}$	$\mathbf{0,37 \pm 0,46}$	$\mathbf{1,25 \pm 1,52}$
CC	$3,90 \pm 4,95$	$3,99 \pm 5,11$	$4,16 \pm 5,34$	$4,40 \pm 5,59$	$5,29 \pm 6,61$	$8,34 \pm 10,38$
RAkEL1	$37,49 \pm 73,2$	$37,72 \pm 73$	$37,76 \pm 74,5$	$38,24 \pm 72,7$	$40,25 \pm 72,1$	$48,39 \pm 81$
RAkEL2	$40,63 \pm 70$	$39,65 \pm 68,4$	$41,34 \pm 72$	$42,21 \pm 72,5$	$46,65 \pm 75,6$	$61,47 \pm 89,2$
MLkNN	$32,42 \pm 57,4$	$31,69 \pm 56,5$	$31,87 \pm 57$	$32,43 \pm 56,9$	$32,28 \pm 57,5$	$34,54 \pm 61,3$
HOMER	$35,34 \pm 61,7$	$35,50 \pm 62,4$	$35,12 \pm 62,8$	$36,07 \pm 62$	$36,08 \pm 62,1$	$39,16 \pm 65,3$
IBLR-ML	$31,33 \pm 55,8$	$32,17 \pm 56,5$	$33,03 \pm 57,9$	$33,73 \pm 62,6$	$34,11 \pm 61,9$	$35,82 \pm 64,6$
CLR	$61,24 \pm 108$	$61,18 \pm 109$	$60,88 \pm 110$	$61,88 \pm 110$	$62,74 \pm 111$	$68,04 \pm 115$
ECC	$39,84 \pm 50,8$	$39,78 \pm 50,6$	$40,28 \pm 51,4$	$42,41 \pm 53,9$	$47,46 \pm 60,0$	$63,00 \pm 78,4$
BR	$8,64 \pm 11$	$8,88 \pm 11,4$	$9,00 \pm 11,4$	$9,42 \pm 12$	$10,64 \pm 13,5$	$14,26 \pm 17,8$
EBR	$92,47 \pm 117$	$92,20 \pm 117$	$94,48 \pm 121$	$96,77 \pm 12$	$103,21 \pm 131$	$121,53 \pm 153$
RF-PCT	$3,35 \pm 4,55$	$3,61 \pm 4,88$	$4,31 \pm 5,72$	$5,84 \pm 7,47$	$9,67 \pm 11,8$	$20,25 \pm 24,3$

Table 11: The average training times of each classifier for each training set size (in seconds).

Classifier	Training complexity	Prediction complexity	Reference
RF-PCT	$\mathcal{O}(N \,.\, q \,.\, n \,.\, m' \,.\, log(n))$	$\mathcal{O}(N \,.\, log(n))$	(Kocev, 2012)
Baseline	$\mathcal{O}(n)$	$\mathcal{O}(1)$	/
LP	$\mathcal{O}(h_M(n, m, 2^q))$	$\mathcal{O}(h'_M(m, 2^q))$	(Zhang and Zhou, 2013)
BR	$\mathcal{O}(q \,.\, h_B(n, m))$	$\mathcal{O}(q \,.\, h'_B(m))$	(Zhang and Zhou, 2013)
CC	$\mathcal{O}(q \,.\, h_B(n, m + q))$	$\mathcal{O}(q \,.\, h'_B(m + q))$	(Zhang and Zhou, 2013)
MLkNN	$\mathcal{O}(n^2 \,.\, m \,+\, q \,.\, n \,.\, k)$	$\mathcal{O}(n \,.\, m \,+\, q \,.\, k)$	(Zhang and Zhou, 2013)
IBLR_ML	$\mathcal{O}(n^2 \,.\, m \,+\, q \,.\, n \,.\, k)$	$\mathcal{O}(n \,.\, m \,+\, q \,.\, k)$	/
RAkEL	$\mathcal{O}(N \,.\, h_M(n, m, 2^k))$	$\mathcal{O}(N \,.\, h'_M(m, 2^k))$	(Zhang and Zhou, 2013)
HOMER	$\mathcal{O}(C(q) \,+\, q)$	$\mathcal{O}(log_k(q))$	(Tsoumakas et al., 2008)
EBR	$\mathcal{O}(N \,.\, q \,.\, h_B(n, m))$	$\mathcal{O}(N \,.\, q \,.\, h'_B(m))$	(Zhang and Zhou, 2013)
ECC	$\mathcal{O}(N \,.\, q \,.\, h_B(n, m + q))$	$\mathcal{O}(N \,.\, q \,.\, h'_B(m + q))$	(Zhang and Zhou, 2013)
CLR	$\mathcal{O}(q^2 \,.\, h_B(n, m))$	$\mathcal{O}(q^2 \,.\, h'_B(m))$	(Zhang and Zhou, 2013)

Table 12: The computational complexities of each classifier for both training and predicting in terms of number of training examples (n), number of features (m) and number of labels (q). N is the number of base learners for the ensemble methods. $C(.)$ is the computational complexity of the balanced clustering algorithm. k could be the number of label subsets, the size of label subsets or the number of neighbours respectively for HOMER, RAkEL and instance-based methods. m' is the number of features selected at each node in RF-PCT. And, $h_B(.)$ (resp. $h_M(.)$) and $h'_B(.)$ (resp. $h'_M(.)$) denote the training and per-instance testing computational complexities of the binary (resp. multi-class) base learner B (resp. M) used in problem transformation approaches.

7. Conclusion and Future work

Integrating multi-label classification in an interactive framework is a promising research area which has recently been stimulated by real-life applications from various domains. In the last decade, numerous multi-label classification approaches have been developed. But a major question is the selection of an algorithm which resists the interactivity constraints.

To the best of our knowledge, this paper presents the first extensive comparative study of multi-label learning algorithms in an interactive setting. We have compared twelve well-established multi-label learning algorithms from three families (problem transformation methods, adaptation methods, ensemble methods) for twelve datasets of different sizes from various domains. The quality of their predictions was evaluated for five complementary multi-label criteria: ranking-based criteria for the labels and the examples (RL and

macro-RL), Accuracy, F_1-score and Balanced Error Rate (BER)). For strengthening our conclusions, we have considered five additional measures from the literature: Coverage, One Error, Average Precision, Hamming Loss and Exact match. Their computation efficiency is basically evaluated by some observed running times and by their theoretical computational complexities for both training and predicting. Our analysis is focused on the first phase of the classification task where only few training examples are available. In practice, this phase is essential to gain user confidence in the interactive system.

Our comparison shows that four classifiers can be distinguished for the prediction quality: RF-PCT (Random Forest of Predictive Clustering Trees), EBR (Ensemble of Binary Relevance), CLR (Calibrated Label Ranking) and MLkNN (Multi-label kNN) with an advantage for the first two ensemble classifiers. Moreover, RF-PCT also competes with the fastest classifiers that obtain poor predictive performances. Consequently, we conclude that RF-PCT, which was already distinguished for the classical multi-label classification (Madjarov et al., 2012), still remains efficient for an interactive multi-label classification.

In the next future, we plan to follow two complementary research directions: (i) improving the best approaches, in particular RF-PCT and CLR, by exploiting the structure of the unclassified data, and (ii) complexifying our experimental protocol. When the training data size is limited, it is commonly argued that the information induced from unclassified data enables learners to significantly improve their predictive performance (Chapelle et al., 2006). We first want to extend our classifier comparison by confronting the best learners of this actual study to existing semi-supervised multi-label learning approaches (e.g. Liu et al. (2006) and Kong et al. (2013)). Our ambition is to better understand the contribution of this added information to develop a more efficient interactive multi-label classification algorithm.

In this study we have identified the classifiers which satisfy the main constraints of any interactive environment. The next step is to analyse their behaviours in a more realistic

framework by simulating user selection/correction and user addition of new labels of interest. We are currently developing a prototype of an interactive learning system for the *Video On Demand* and we will soon conduct a subjective user evaluation of the system with each of the top classifiers.

Acknowledgement

We thank anonymous reviewers whose critical feedback and valuable suggestions have helped to improve and clarify the manuscript.

References

Amershi, S. (2011). Designing for effective end-user interaction with machine learning. In *Proceedings of the 24th annual ACM symposium adjunct on User interface software and technology*, pages 47–50. ACM.

Amershi, S., Cakmak, M., Knox, W. B., and Kulesza, T. (2015). Power to the people: The role of humans in interactive machine learning. *AI Magazine (Accepted and in press)*.

Amershi, S., Fogarty, J., and Weld, D. (2012). Regroup: Interactive machine learning for on-demand group creation in social networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 21–30. ACM.

Amershi, S., Lee, B., Kapoor, A., Mahajan, R., and Christian, B. (2011). Cuet: human-guided fast and accurate network alarm triage. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 157–166. ACM.

Basu, S., Fisher, D., Drucker, S. M., and Lu, H. (2010). Assisting users with clustering tasks by combining metric learning and classification. In *AAAI*.

Bilenko, M., Basu, S., and Mooney, R. J. (2004). Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 11. ACM.

Boutell, M. R., Luo, J., Shen, X., and Brown, C. M. (2004). Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771.

Briggs, F., Huang, Y., Raich, R., Eftaxias, K., Lei, Z., Cukierski, W., Hadley, S. F., Hadley, A., Betts, M., Fern, X. Z., et al. (2013). The 9th annual mlsp competition: New methods for acoustic classification of

multiple simultaneous bird species in a noisy environment. In *Machine Learning for Signal Processing (MLSP), 2013 IEEE International Workshop on*, pages 1–8. IEEE.

Chapelle, O., Schölkopf, B., Zien, A., et al. (2006). *Semi-supervised learning*, volume 2. MIT press Cambridge.

Chen, Y.-W. and Lin, C.-J. (2006). Combining svms with various feature selection strategies. In *Feature extraction*, pages 315–324. Springer.

Cheng, W. and Hüllermeier, E. (2009). Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2-3):211–225.

Cheng, W., Hüllermeier, E., and Dembczynski, K. J. (2010). Graded multilabel classification: The ordinal case. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 223–230.

Clare, A. and King, R. D. (2001). Knowledge discovery in multi-label phenotype data. In *Principles of data mining and knowledge discovery*, pages 42–53. Springer.

Dabrowski, J. R. and Munson, E. V. (2001). Is 100 milliseconds too fast? In *CHI'01 Extended Abstracts on Human Factors in Computing Systems*, pages 317–318. ACM.

Diplaris, S., Tsoumakas, G., Mitkas, P. A., and Vlahavas, I. (2005). Protein classification with multiple algorithms. In *Advances in Informatics*, pages 448–456. Springer.

Drucker, S. M., Fisher, D., and Basu, S. (2011). Helping users sort faster with adaptive machine learning recommendations. In *Human-Computer Interaction–INTERACT 2011*, pages 187–203. Springer.

Elisseeff, A. and Weston, J. (2001). A kernel method for multi-labelled classification. In *Advances in neural information processing systems*, pages 681–687.

Fails, J. A. and Olsen Jr, D. R. (2003). Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 39–45. ACM.

Fiebrink, R., Trueman, D., and Cook, P. R. (2009). A metainstrument for interactive, on-the-fly machine learning. In *Proc. NIME*, volume 2, page 3.

Fogarty, J., Tan, D., Kapoor, A., and Winder, S. (2008). Cueflik: interactive concept learning in image search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 29–38. ACM.

Fürnkranz, J., Hüllermeier, E., Mencía, E. L., and Brinker, K. (2008). Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153.

Ghamrawi, N. and McCallum, A. (2005). Collective multi-label classification. In *Proceedings of the 14th*

ACM international conference on Information and knowledge management, pages 195–200. ACM.

Godbole, S. and Sarawagi, S. (2004). Discriminative methods for multi-labeled classification. In *Advances in Knowledge Discovery and Data Mining*, pages 22–30. Springer.

Hüllermeier, E., Fürnkranz, J., Cheng, W., and Brinker, K. (2008). Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16):1897–1916.

Katakis, I., Tsoumakas, G., and Vlahavas, I. (2008). Multilabel text classification for automated tag suggestion. In *Proceedings of the ECML/PKDD*.

Kocev, D. (2012). Ensembles for predicting structured outputs. *Informatica: An International Journal of Computing and Informatics*, 36(1):113–114.

Kocev, D., Vens, C., Struyf, J., and Džeroski, S. (2007). *Ensembles of multi-objective decision trees*. Springer.

Kong, X., Ng, M. K., and Zhou, Z.-H. (2013). Transductive multilabel learning via label set propagation. *Knowledge and Data Engineering, IEEE Transactions on*, 25(3):704–719.

Li, T., Zhang, C., and Zhu, S. (2006). Empirical studies on multi-label classification. In *IcTAI*, volume 6, pages 86–92.

Lintott, C. J., Schawinski, K., Slosar, A., Land, K., Bamford, S., Thomas, D., Raddick, M. J., Nichol, R. C., Szalay, A., Andreescu, D., et al. (2008). Galaxy zoo: morphologies derived from visual inspection of galaxies from the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society*, 389(3):1179–1189.

Liu, Y., Jin, R., and Yang, L. (2006). Semi-supervised multi-label learning by constrained non-negative matrix factorization. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 421. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Lo, H.-Y., Wang, J.-C., Wang, H.-M., and Lin, S.-D. (2011). Cost-sensitive multi-label learning for audio tag annotation and retrieval. *Multimedia, IEEE Transactions on*, 13(3):518–529.

Madjarov, G., Kocev, D., Gjorgjevikj, D., and Džeroski, S. (2012). An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, 45(9):3084–3104.

Nasierding, G. and Kouzani, A. Z. (2012). Comparative evaluation of multi-label classification methods. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*, pages 679–683. IEEE.

Ozonat, K. and Young, D. (2009). Towards a universal marketplace over the web: Statistical multi-label classification of service provider forms with simulated annealing. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1295–1304. ACM.

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers*. Citeseer.

Porter, R., Theiler, J., and Hush, D. (2013). Interactive machine learning in data exploitation. *Computing in Science & Engineering*, 15(5):12–20.

Quinlan, J. R. (1993). *C4. 5: programs for machine learning*, volume 1. Morgan kaufmann.

Rak, R., Kurgan, L., and Reformat, M. (2005). Multi-label associative classification of medical documents from medline. In *Machine Learning and Applications, 2005. Proceedings. Fourth International Conference on*, pages 8–pp. IEEE.

Read, J. (2010). *Scalable multi-label classification*. PhD thesis, University of Waikato.

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2011). Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359.

Ritter, A. and Basu, S. (2009). Learning to generalize for complex selection tasks. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 167–176. ACM.

Salton, G. and Buckley, C. (1997). Improving retrieval performance by relevance feedback. *Readings in information retrieval*, 24(5).

Schapire, R. E. and Singer, Y. (2000). Boostexter: A boosting-based system for text categorization. *Machine learning*, 39(2-3):135–168.

Simon, H. A. (1955). A behavioral model of rational choice. *The quarterly journal of economics*, pages 99–118.

Snoek, C. G., Worring, M., Van Gemert, J. C., Geusebroek, J.-M., and Smeulders, A. W. (2006). The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 421–430. ACM.

Sorower, M. S. (2010). A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*.

Spyromitros, E., Tsoumakas, G., and Vlahavas, I. (2008). An empirical study of lazy multilabel classification algorithms. In *Artificial Intelligence: Theories, Models and Applications*, pages 401–406. Springer.

Srivastava, A. N. and Zane-Ulman, B. (2005). Discovering recurring anomalies in text reports regarding complex space systems. In *Aerospace Conference, 2005 IEEE*, pages 3853–3862. IEEE.

Stumpf, S., Rajaram, V., Li, L., Burnett, M., Dietterich, T., Sullivan, E., Drummond, R., and Herlocker, J. (2007). Toward harnessing user feedback for machine learning. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 82–91. ACM.

Tawiah, C. A. and Sheng, V. S. (2013). A study on multi-label classification. In *Advances in Data Mining. Applications and Theoretical Aspects*, pages 137–150. Springer.

Trohidis, K., Tsoumakas, G., Kalliris, G., and Vlahavas, I. P. (2008). Multi-label classification of music into emotions. In *ISMIR*, volume 8, pages 325–330.

Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13.

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2008). Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, pages 30–44.

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2010). Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer.

Tsoumakas, G. and Vlahavas, I. (2007). Random k-labelsets: An ensemble method for multilabel classification. In *Machine Learning: ECML 2007*, pages 406–417. Springer.

Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S., et al. (2001). Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584.

Ware, M., Frank, E., Holmes, G., Hall, M., and Witten, I. H. (2001). Interactive machine learning: letting users build classifiers. *International Journal of Human-Computer Studies*, 55(3):281–292.

Yu, K., Yu, S., and Tresp, V. (2005). Multi-label informed latent semantic indexing. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 258–265. ACM.

Zhang, M. and Zhou, Z. (2013). A review on multi-label learning algorithms.

Zhang, M.-L. and Zhou, Z.-H. (2007). Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048.

Appendix 1

The detailed results obtained for three representative training data sizes (4, 16 and 64 examples) for the main quality criteria (Ranking Loss and macro-averaged Ranking Loss) are presented below.

	Baseline	LP	CC	RAkEL1	RAkEL2	MLkNN	HOMER	IBLR-ML	CLR	ECC	BR	EBR	RF-PCT
Emotions	0,45	**0,44**	0,46	0,45	0,45	0,50	0,45	0,50	0,48	0,48	0,45	0,47	0,46
Yeast	0,37	0,32	0,32	0,30	0,30	0,30	0,36	0,32	**0,28**	0,29	0,31	**0,28**	**0,28**
Scene	0,50	**0,49**	0,50	0,50	0,50	0,50	0,51	0,50	0,50	0,50	0,50	0,50	**0,49**
Birds	0,26	0,25	0,24	0,25	0,25	0,24	0,26	0,24	0,24	0,25	0,24	0,24	**0,23**
Slashdot	0,57	0,49	0,49	0,52	0,56	0,49	0,59	0,49	**0,48**	0,49	0,49	**0,48**	**0,48**
IMDB	0,55	0,41	0,38	0,45	0,49	0,36	0,55	0,36	0,36	0,39	0,38	**0,35**	**0,35**
Genbase	0,29	**0,27**	0,28	0,29	0,32	0,29	0,42	0,29	0,28	**0,27**	**0,27**	**0,27**	**0,27**
Arts	0,40	0,40	0,39	0,40	0,42	0,38	0,46	**0,37**	**0,37**	0,39	0,39	**0,37**	**0,37**
Business	0,11	0,12	0,10	0,11	0,11	0,09	0,20	0,09	**0,08**	0,09	0,10	0,09	**0,08**
Health	0,28	0,26	0,24	0,26	0,27	0,22	0,34	0,22	**0,21**	0,24	0,25	0,22	**0,21**
Computers	0,23	0,25	0,27	0,25	0,27	0,23	0,33	0,23	**0,22**	0,23	0,27	0,23	0,23
TMC	0,42	0,33	0,31	0,32	0,35	**0,28**	0,42	**0,28**	**0,28**	0,29	0,31	**0,28**	**0,28**

Table .13: The average performances of each classifier for all training sets of size 4 of each dataset in terms of the Ranking Loss criterion

	Baseline	LP	CC	RAkEL1	RAkEL2	MLkNN	HOMER	IBLR-ML	CLR	ECC	BR	EBR	RF-PCT
Emotions	0,46	0,38	0,38	0,34	0,34	0,38	0,39	0,40	0,33	0,31	0,37	0,31	**0,29**
Yeast	0,31	0,33	0,40	0,29	0,29	0,25	0,35	0,34	0,26	0,27	0,35	0,27	**0,24**
Scene	0,49	0,44	0,41	0,39	0,39	0,36	0,45	0,40	0,44	0,39	0,41	0,37	**0,34**
Birds	0,26	0,24	0,23	0,23	0,23	**0,20**	0,26	0,24	0,21	0,22	0,23	0,21	**0,20**
Slashdot	0,53	0,40	0,37	0,48	0,53	0,36	0,55	0,37	0,35	0,51	0,37	0,35	**0,34**
IMDB	0,46	0,39	0,29	0,42	0,50	**0,24**	0,55	0,27	**0,24**	0,41	0,28	0,25	**0,24**
Genbase	0,27	0,15	0,16	0,17	0,18	0,15	0,22	0,19	0,21	**0,11**	0,16	0,14	0,18
Arts	0,39	0,41	0,36	0,38	0,40	**0,27**	0,51	0,29	0,28	0,38	0,36	0,30	**0,27**
Business	0,10	0,10	0,10	0,10	0,10	**0,07**	0,17	0,08	**0,07**	0,09	0,10	0,08	**0,07**
Health	0,25	0,24	0,22	0,23	0,24	**0,15**	0,39	0,16	**0,15**	0,21	0,22	0,16	**0,15**
Computers	0,21	0,22	0,22	0,22	0,23	**0,17**	0,35	0,18	**0,17**	0,21	0,23	0,19	**0,17**
TMC	0,38	0,32	0,31	0,29	0,30	0,22	0,41	0,25	0,22	0,26	0,32	0,24	**0,21**

Table .14: The average performances of each classifier for all training sets of size 16 of each dataset in terms of the Ranking Loss criterion

	Baseline	LP	CC	RAkEL1	RAkEL2	MLkNN	HOMER	IBLR-ML	CLR	ECC	BR	EBR	RF-PCT
Emotions	0,46	0,33	0,35	0,25	0,25	0,29	0,34	0,28	0,25	0,23	0,32	0,23	**0,19**
Yeast	0,28	0,31	0,42	0,25	0,25	0,21	0,33	0,26	0,23	0,23	0,34	0,23	**0,20**
Scene	0,49	0,30	0,33	0,24	0,23	0,25	0,35	0,25	0,21	0,22	0,34	0,22	**0,14**
Birds	0,26	0,20	0,21	0,21	0,21	0,16	0,24	0,22	0,16	0,18	0,21	0,17	**0,15**
Slashdot	0,52	0,34	0,26	0,41	0,46	0,27	0,56	0,28	0,24	0,41	0,26	0,24	**0,22**
IMDB	0,41	0,40	0,25	0,39	0,48	**0,20**	0,57	0,23	**0,20**	0,39	0,25	0,22	0,21
Genbase	0,27	**0,03**	0,04	0,05	0,05	**0,03**	0,05	0,04	0,06	**0,03**	**0,03**	**0,03**	**0,03**
Arts	0,39	0,36	0,28	0,34	0,36	0,22	0,49	0,26	0,22	0,31	0,27	0,23	**0,20**
Business	0,10	0,09	0,08	0,09	0,09	**0,06**	0,16	0,07	**0,06**	0,08	0,09	**0,06**	**0,06**
Health	0,25	0,21	0,17	0,20	0,20	0,12	0,37	0,15	0,12	0,18	0,18	0,13	**0,11**
Computers	0,21	0,21	0,18	0,20	0,21	**0,13**	0,38	0,15	**0,13**	0,19	0,20	0,14	**0,13**
TMC	0,36	0,29	0,29	0,24	0,25	0,18	0,38	0,22	0,17	0,21	0,29	0,18	**0,15**

Table .15: The average performances of each classifier for all training sets of size 64 of each dataset in terms of the Ranking Loss criterion

	Baseline	LP	CC	RAkEL1	RAkEL2	MLkNN	HOMER	IBLR-ML	CLR	ECC	BR	EBR	RF-PCT
Emotions	0,50	0,45	0,49	0,45	0,45	0,50	0,47	0,52	0,48	0,48	0,48	0,48	**0,40**
Yeast	**0,48**	0,49	**0,48**	0,49	0,49	0,49	0,49	0,50	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**
Scene	0,50	0,49	0,50	0,49	0,49	0,49	0,50	0,51	0,50	0,50	0,50	0,50	**0,48**
Birds	0,49	0,49	0,49	0,49	0,49	0,49	0,49	0,50	0,49	0,49	0,49	0,49	**0,48**
Slashdot	0,46	0,46	0,46	0,46	0,46	0,46	0,47	0,47	0,46	0,46	0,46	0,46	**0,44**
IMDB	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**	0,49	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**
Genbase	0,48	0,47	0,48	0,47	0,47	0,46	0,47	0,52	0,48	0,44	0,48	0,44	**0,41**
Arts	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**	0,49	0,49	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**
Business	**0,44**	**0,44**	**0,44**	**0,44**	**0,44**	**0,44**	0,45	0,45	**0,44**	**0,44**	**0,44**	**0,44**	**0,44**
Health	0,49	0,49	0,49	0,49	0,49	0,49	0,49	0,49	0,49	0,49	0,49	0,49	**0,48**
Computers	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**	**0,50**
TMC	0,51	0,51	0,51	0,51	0,51	0,51	0,50	0,51	0,51	0,51	0,51	0,51	**0,49**

Table .16: The average performances of each classifier for all training sets of size 4 of each dataset in terms of the macro-averaged Ranking Loss criterion

	Baseline	LP	CC	RAkEL1	RAkEL2	MLkNN	HOMER	IBLR-ML	CLR	ECC	BR	EBR	RF-PCT
Emotions	0,50	0,41	0,41	0,34	0,34	0,41	0,41	0,41	0,32	0,30	0,40	0,30	**0,23**
Yeast	0,48	0,48	0,47	0,48	0,47	0,48	0,49	0,49	0,47	0,46	0,47	0,46	**0,45**
Scene	0,50	0,43	0,42	0,37	0,37	0,37	0,45	0,38	0,41	0,35	0,42	0,35	**0,25**
Birds	0,49	0,46	0,48	0,45	0,46	0,46	0,43	0,50	0,47	0,44	0,48	0,44	**0,40**
Slashdot	0,46	0,43	0,45	0,44	0,45	0,46	0,45	0,47	0,44	0,44	0,45	0,44	**0,39**
IMDB	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**	0,50	0,49	**0,48**	**0,48**	**0,48**	**0,48**	**0,48**
Genbase	0,48	0,35	0,40	0,38	0,39	0,32	0,32	0,40	0,40	0,32	0,40	0,32	**0,29**
Arts	0,48	0,48	0,48	0,48	0,48	0,48	0,50	0,50	0,48	0,48	0,48	0,48	**0,46**
Business	0,44	0,44	0,44	0,44	0,44	0,44	0,44	0,46	0,44	0,44	0,44	0,44	**0,42**
Health	0,49	0,48	0,48	0,47	0,48	0,48	0,48	0,49	0,48	0,47	0,48	0,47	**0,45**
Computers	0,50	0,50	0,50	0,50	0,50	0,50	0,49	0,50	0,50	0,50	0,50	0,50	**0,48**
TMC	0,51	0,50	0,50	0,48	0,48	0,50	0,50	0,50	0,49	0,49	0,50	0,49	**0,43**

Table .17: The average performances of each classifier for all training sets of size 16 of each dataset in terms of the macro-averaged Ranking Loss criterion

	Baseline	LP	CC	RAkEL1	RAkEL2	MLkNN	HOMER	IBLR-ML	CLR	ECC	BR	EBR	RF-PCT
Emotions	0,50	0,35	0,37	0,26	0,25	0,32	0,37	0,29	0,25	0,23	0,34	0,23	**0,18**
Yeast	0,48	0,46	0,46	0,44	0,43	0,45	0,47	0,46	0,43	0,42	0,46	0,42	**0,39**
Scene	0,50	0,31	0,34	0,23	0,23	0,28	0,35	0,25	0,19	0,22	0,34	0,22	**0,10**
Birds	0,49	0,39	0,45	0,39	0,40	0,40	0,46	0,50	0,41	0,36	0,45	0,36	**0,26**
Slashdot	0,46	0,40	0,43	0,40	0,41	0,46	0,44	0,46	0,38	0,40	0,43	0,40	**0,32**
IMDB	0,48	0,48	0,48	0,48	0,48	0,48	0,49	0,49	0,48	0,48	0,48	0,48	**0,47**
Genbase	0,48	0,22	0,27	0,25	0,25	0,20	0,22	0,22	0,25	0,19	0,26	0,19	**0,17**
Arts	0,48	0,47	0,47	0,46	0,46	0,48	0,48	0,49	0,45	0,45	0,47	0,45	**0,42**
Business	0,44	0,43	0,44	0,43	0,43	0,44	0,46	0,47	0,43	0,42	0,44	0,42	**0,37**
Health	0,49	0,46	0,47	0,44	0,44	0,48	0,47	0,49	0,44	0,43	0,47	0,43	**0,38**
Computers	0,50	0,49	0,50	0,48	0,48	0,50	0,49	0,49	0,49	0,48	0,49	0,48	**0,43**
TMC	0,51	0,47	0,46	0,43	0,43	0,49	0,48	0,47	0,39	0,42	0,46	0,42	**0,29**

Table .18: The average performances of each classifier for all training sets of size 64 of each dataset in terms of the macro-averaged Ranking Loss criterion

Appendix 2

Definitions of the additional quality criteria and the obtained results are given below.

1. Additional quality criteria

The ***Hamming Loss*** evaluates the number of misclassified labels for an example x_i:

$$Hamming - Loss = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} |y_i \triangle \hat{y}_i|$$

where $\triangle$ is the symmetric difference between its ground truth and predicted label sets (y_i and $\hat{y}_i$).

The ***Exact match*** is a very strict criterion as it harshly punishes the model predictions. It requires an exact match between the ground truth label set y_i and the predicted label set $\hat{y}_i$ for an example x_i:

$$Exact\ match = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} I[y_i = \hat{y}_i]$$

where I(true)=1 and I(false)=0.

The ***Coverage*** evaluates how many steps are required, on average, to go down the ranked label list so as to cover all the relevant labels of an example x_i:

$$Coverage = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \max_{\lambda_j \in y_i^+} r_i(\lambda_j) - 1$$

The ***One error*** evaluates how many times the top-ranked label is not relevant for an example x_i;

$$One - Error = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \left| \underset{\lambda_j \in y_i^+}{\operatorname{argmax}} \hat{y}_i^j \notin y_i \right|$$

The ***Average precision*** evaluates the average fraction of labels $\lambda_k \in y_i^+$ ranked above a label $\lambda_j \in y_i^+$ for an example x_i:

$$AveragePrecision = \frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}\frac{1}{|y_i^+|}\sum_{\lambda_j \in y_i^+}\frac{|w_i|}{r_i(\lambda_j)}$$

where $w_i = \{\lambda_k | r_i^k \leq r_i^j, \lambda_k \in y_i^+\}$

2. Experimental results

	2	4	8	16	32	64
MLkNN	$0,73 \pm 0,20$	$0,68 \pm 0,23$	$\mathbf{0,60 \pm 0,22}$	$\mathbf{0,54 \pm 0,23}$	$\mathbf{0,50 \pm 0,25}$	$0,48 \pm 0,26$
CLR	$\mathbf{0,69 \pm 0,22}$	$0,66 \pm 0,23$	$0,62 \pm 0,23$	$0,58 \pm 0,22$	$0,52 \pm 0,22$	$0,48 \pm 0,23$
EBR	$0,72 \pm 0,19$	$0,67 \pm 0,21$	$\mathbf{0,60 \pm 0,22}$	$0,55 \pm 0,23$	$0,51 \pm 0,24$	$0,47 \pm 0,24$
RF-PCT	$0,72 \pm 0,19$	$\mathbf{0,65 \pm 0,22}$	$0,61 \pm 0,23$	$0,56 \pm 0,22$	$\mathbf{0,50 \pm 0,21}$	$\mathbf{0,45 \pm 0,23}$

Table .19: The average performances of the *top*(4) classifiers for each training set size for the One-error criterion.

	2	4	8	16	32	64
MLkNN	$\mathbf{0,19 \pm 0,12}$	$0,20 \pm 0,12$	$0,17 \pm 0,11$	$0,15 \pm 0,10$	$0,14 \pm 0,09$	$0,13 \pm 0,08$
CLR	$\mathbf{0,19 \pm 0,12}$	$0,18 \pm 0,12$	$0,16 \pm 0,11$	$0,16 \pm 0,10$	$0,14 \pm 0,09$	$0,13 \pm 0,08$
EBR	$\mathbf{0,19 \pm 0,12}$	$0,20 \pm 0,11$	$0,16 \pm 0,10$	$0,14 \pm 0,09$	$0,13 \pm 0,08$	$0,12 \pm 0,08$
RF-PCT	$\mathbf{0,19 \pm 0,12}$	$\mathbf{0,16 \pm 0,11}$	$\mathbf{0,14 \pm 0,10}$	$\mathbf{0,13 \pm 0,08}$	$\mathbf{0,11 \pm 0,07}$	$\mathbf{0,10 \pm 0,07}$

Table .20: The average performances of the *top*(4) classifiers for each training set size for the Hamming loss criterion.

	2	4	8	16	32	64
MLkNN	$0,42 \pm 0,16$	$0,45 \pm 0,16$	$\mathbf{0,52 \pm 0,15}$	$\mathbf{0,57 \pm 0,16}$	$0,60 \pm 0,17$	$0,63 \pm 0,18$
CLR	$\mathbf{0,44 \pm 0,16}$	$\mathbf{0,47 \pm 0,17}$	$0,50 \pm 0,16$	$0,54 \pm 0,15$	$0,59 \pm 0,15$	$0,63 \pm 0,16$
EBR	$0,42 \pm 0,16$	$0,46 \pm 0,16$	$\mathbf{0,52 \pm 0,16}$	$0,56 \pm 0,16$	$0,60 \pm 0,17$	$0,63 \pm 0,17$
RF-PCT	$0,43 \pm 0,16$	$\mathbf{0,47 \pm 0,16}$	$\mathbf{0,52 \pm 0,16}$	$0,56 \pm 0,15$	$\mathbf{0,62 \pm 0,15}$	$\mathbf{0,66 \pm 0,16}$

Table .21: The average performances of the *top*(4) classifiers for each training set size for the Average precision criterion.

	2	4	8	16	32	64
MLkNN	$0,04 \pm 0,03$	$0,04 \pm 0,03$	$0,06 \pm 0,05$	$0,06 \pm 0,05$	$0,07 \pm 0,05$	$0,09 \pm 0,09$
CLR	**0,05 ± 0,03**	$0,04 \pm 0,03$	$0,05 \pm 0,04$	$0,04 \pm 0,04$	$0,06 \pm 0,04$	$0,08 \pm 0,05$
EBR	$0,04 \pm 0,03$	$0,04 \pm 0,04$	$0,10 \pm 0,07$	$0,13 \pm 0,11$	$0,16 \pm 0,12$	$0,20 \pm 0,13$
RF-PCT	**0,05 ± 0,03**	**0,10 ± 0,08**	**0,13 ± 0,09**	**0,17 ± 0,11**	**0,21 ± 0,14**	**0,26 ± 0,18**

Table .22: The average performances of the *top*(4) classifiers for each training set size for the Exact match criterion.

	2	4	8	16	32	64
MLkNN	$8,32 \pm 3,92$	$7,74 \pm 3,35$	$6,89 \pm 2,81$	$6,04 \pm 2,48$	$5,33 \pm 2,36$	$4,79 \pm 2,32$
CLR	**8,26 ± 3,93**	**7,61 ± 3,34**	$6,89 \pm 2,76$	$6,13 \pm 2,42$	$5,41 \pm 2,29$	$4,80 \pm 2,31$
EBR	$8,30 \pm 3,92$	$7,66 \pm 3,31$	$6,91 \pm 2,85$	$6,17 \pm 2,65$	$5,46 \pm 2,57$	$4,91 \pm 2,54$
RF-PCT	$8,31 \pm 3,92$	$7,62 \pm 3,32$	**6,81 ± 2,78**	**5,94 ± 2,47**	**5,12 ± 2,36**	**4,48 ± 2,38**

Table .23: The average performances of the *top*(4) classifiers for each training set size for the Coverage criterion.

Appendix 3: Critical diagrams

The critical diagram represents a projection of the classifier average ranks on an enumerated axis. The classifiers are placed from left (best) to right (worst) and a bold line connects those whose average ranks do not differ significantly (for the significance level 0.05).

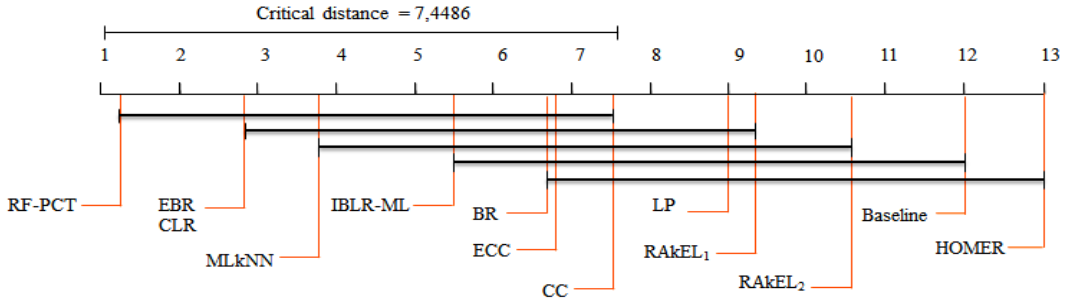


Figure .2: The critical diagram for the Ranking loss criterion for all training data sizes.

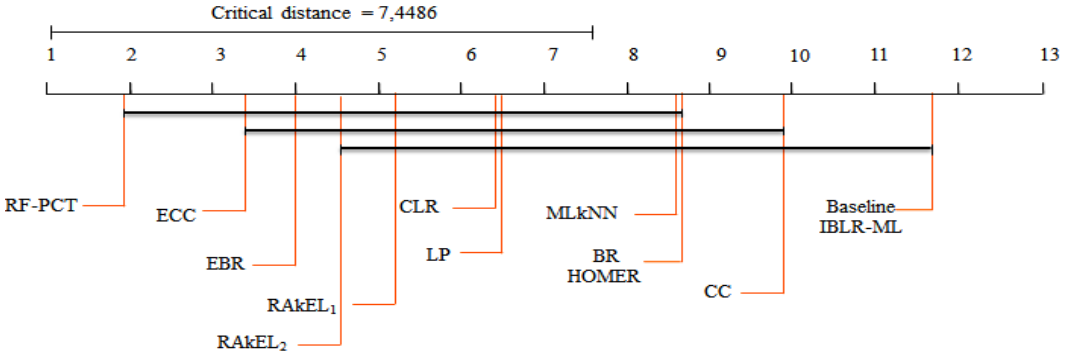
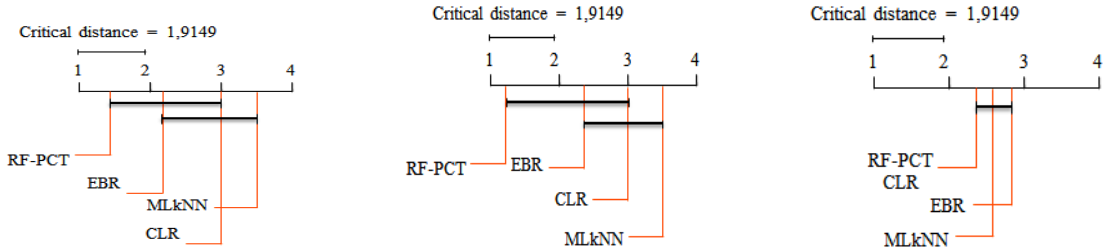


Figure .3: The critical diagram for the macro-averaged Ranking loss criterion for all training data sizes.



(a) F_1-score (b) Accuracy (c) BER

Figure .4: The critical diagrams for the Bipartition-based criteria: (a) F_1-score, (b) Accuracy and (c) BER for all training data sizes.