

A New Approach for Improving Accuracy of Multi Label Stream Data

Kunal Shah¹ Swati Patel²

^{1,2}Department of Information Technology

^{1,2}L.D. College of Engineering, Ahmedabad, Gujarat, 380015

Abstract— Many real world problems involve data which can be considered as multi-label data streams. Efficient methods exist for multi-label classification in non-streaming scenarios. However, learning in evolving streaming scenarios is more challenging, as the learners must be able to adapt to change using limited time and memory. Classification is used to predict class of unseen instance as accurate as possible. Multi label classification is a variant of single label classification where set of labels associated with single instance. Multi label classification is used by modern applications, such as text classification, functional genomics, image classification, music categorization etc. This paper introduces the task of multi-label classification, methods for multi-label classification and evolution measure for multi-label classification. Also done comparative analysis of multi label classification methods on the basis of theoretical study and then on the basis of simulation done on various data sets.

Keywords: MLSC, Threshold Value, Demerits of Binary Relevance method

I. INTRODUCTION

Real-time analysis of data streams is becoming a key area of data mining research as the number of applications demanding such processing increases. Nowadays, data is generated at an increasing rate from sensor applications, measurements in network monitoring and traffic management, log records or click-streams in web exploring, manufacturing processes, call detail records, email, blogging, twitter posts, and other sources[1,2].

In the traditional supervised classification task, each example is associated with a single class label. A classifier learns to associate each new unseen example with exactly one of these known class labels. When each example may be associated with multiple labels, then this is called multi-label classification. Hence multi-label classification is simply the classification task where each example may be associated with multiple labels.

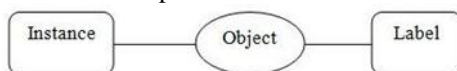


Fig 1: Single Label Classification

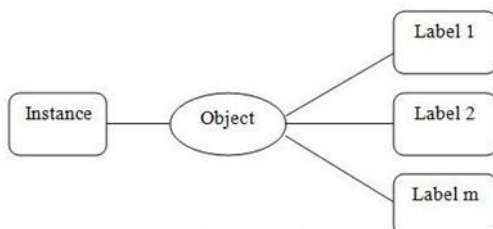


Fig 2: Multi Label Classification

Fig. 1 Single Label Classification, Fig. 2: Multi Label Classification

A classification task usually involves separating data into training and testing sets. Each instance in the training set contains one class label and several attributes.

The goal of classifier is to produce a model which predicts label of the test data given only the test data attributes. In classification problems, each instance of a dataset is associated with just one class label that is single label classification. (As shown in fig. 1)

However, there are many classification tasks where each instance can be associated with one or more class labels. This group of problems represents an area known as Multi-Label Classification. (As shown in fig. 2) Multi-label classification methods are increasingly required by modern applications, such as text classification, gene functionality, music categorization and semantic scene classification. The number of class labels is predicted for each instance[3,4].

Multi-label stream classification (MLSC) has emerged recently as an extension to conventional stream classification in response to applications where arriving data instances can or must acquire more than one label. This typically happens either because the labels are orthogonal or because it is not practical to define labels that are completely distinct and intuitive at the same time. Orthogonal labels are encountered e.g. in the categorization of incoming mails or enterprise document: such instances may be relevant to a thematic label, as well as to a label concerning confidentiality. Such classes are a priori orthogonal, but correlations may be encountered.

Multi-label classification of static data enjoys increased attention in recent years. An overview of the domain[5], including discussion of applications. Solutions for static data do not readily transfer to stream scenarios though, since they assume static concepts and availability of all data for learning. On the other hand, research on single-label stream classification has contributed several powerful algorithms for an overview that exhibit good predictive power and fast adaptation to concept drift. In principle, drift adaptation encompasses monitoring the distribution of positive and negative examples within some window, and discarding old examples when the most recent data indicate a change in the distribution. Some of these solutions have been extended to the multi-label case, without however explicitly addressing a number of challenges that are specific to multi-label streams. A multi-label data stream contains separates multiple targets (concepts), and it is impractical to assume that all of them will start drifting simultaneously or at the same rate. Essentially, each concept is likely to exhibit its own drift pattern. Another challenge of multi-label data is the class imbalance problem: each label has usually more negative than positive examples, but still some labels have much more positive examples than others. If a single window is used, it is expected that some labels will have enough examples for learning the positive class, but many labels may have little or even no positive examples.

We present a new approach that deals with the above challenges of MLSC. Our new Multiple Windows with Buffer (MW-Buffer) method first create a buffer and then maintains two fixed-size windows per label, one for

positive and one for negative examples. This is accomplished in a space-efficient way through instance-sharing between windows. In addition, we present a time-efficient instantiation of this method using Naive Bayes as the base classifier for each label[5,26].

The rest of the paper is organized as follows, in section 2; we discuss existing work on multi-label stream classification and further relevant literature. Section 3 presents our contributions. It is followed by the empirical evaluation in section 4. The last section concludes our study.

II. RELATED WORK

One of the first methods for MLSC is[10,11]. It assumes that stream instances arrive in chunks of size S and builds an ensemble of K classifiers, on K successive chunks. To deal with concept drift, every S example the oldest model is replaced by a model built on the latest chunk. The authors used stacked binary relevance to learn from each chunk, but the method could be coupled with any batch multi-label learner.

Another work dealing with multi-label stream present a framework for generating synthetic multi-label data streams along with a novel MLSC method[14,17] based on the Hoeffding Tree[21], a popular decision tree classifier for single-label streams. Their method extends the Hoeffding Tree by using a multi-label definition of entropy and by training multi-label classifiers at the leaves of the tree. However, it does not offer drift adaptation, hence is not suitable for classifying evolving multi-label streams. The aforementioned approaches try to tackle MLSC by combining existing stream and multi-label classification methods but they do not deal explicitly with the special characteristics of a multi-label stream such as independent concept drift for each label (or group of labels) and skewness in the distribution of positive and negative examples for most of the labels[26].

In single-label data streams, most approaches assume balanced distributions of positive and negative examples. This method processes the stream instances in batches. It tries to build balanced training sets as follows: all positive examples are kept, while the negative examples of the latest chunk are undersampled, and organized (together with the positive ones) into multiple disjoint samples. Then, an ensemble of classifiers is trained, which is completely rebuilt upon the arrival of the latest chunk. Ensemble re-learning is computationally expensive, though. Moreover, retaining all positive examples may 1) prevent the learner from adapting properly to drift and 2) prohibitively increase re-training time. We follow a similar strategy to deal with label skewness in multi-label streams but we impose a limit in the number of positive examples that we keep for each label. This way we overcome the aforementioned disadvantages of the method of and make it more suitable for multi-label streams where each label has each own degree of skewness.

Furthermore, our method is instance incremental and thus allows faster adaptation to drift and avoids the computational overhead of rebuilding the model from scratch[26,27].

In Multiple Window Approach, there are two windows one for positive and one for negative examples. So in this method there are so many negative examples as

compared to positive examples. So it undersamples the positive and oversamples the negative examples. So positive to negative label ratio is very less.

III. OUR CONTRIBUTIONS

A. Multiple Windows with Buffer Approach:

Our approach starts with the buffer approach in that buffer first store all the labels values after that it follows the moving-window approach. As its name implies, this idea is about maintaining a classifier that is trained from a moving window of recent examples[26].

A couple of issues can arise if we attempt to apply this idea to multi-label data: a) each label constitutes a different learning problem, including a different rate of concept drift. b) The distribution of positive and negative examples of for most labels will be skewed, and the negative example are expected to dominate, so that the few positive examples will be insufficient for learning the positive class for some of the labels. An option here is to increase the size of the window, allowing a sufficient number of positive examples for all labels. However, this would increase the probability of concept drift occurrence in the window[26,31].

To deal with the above issues we propose the following approach. We first create a buffer for storing all the positive and negative label values. After that each label is associate with two fixed size instance windows, one for positive and one for negative examples.

First we store all the negative and positive examples in buffer after that we set Threshold Value (T.V.) of that buffer after that if negative to positive examples are below or positive to negative examples are above than Threshold Value at that time we simply transfer that directly to window and continue same procedure on next set of examples, but if negative to positive examples are above or positive to negative examples are below then the Threshold Value (T.V.) at that time simple hold all the examples in buffer and cannot put all in window and wait till the next opposite case means same number of positive to negative examples occur in buffer at that time we simply put all these examples in window and continue same procedure on next set of examples.

Compare to a Multiple Window (MW) approach that would use only multiple window, our approach effectively oversamples the positive and undersamples the negative examples for all labels. The oversampling is achieved by adding the most recent positive examples that appear prior to that window and the undersampling by retaining only the most recent negative examples. Fig 3 contrasts the two approaches.

Our technique reduces the high variance caused by the insufficient positive examples available to a classifier operating in a multiple window, leading to reduced classification error. In the case of concept drift, the bias may increase by the introduction of old positive examples caused by oversampling. However, this increase is expected to be small because the negative examples are expected to always be current.

We follow the binary relevance (BR) approach, since we transform the multi-label problem into multiple binary problems and tackle each problem independently. BR

has been criticized in the past for ignoring potential underlying label correlations, but, in the meanwhile, there are BR-based methods that overcome this limitation. We use the independent modelling of BR, because it allows us to effectively handle the expected differences in frequency and concept drift between the labels. BR offer also a number of further advantages: a) it can be combined with any binary classification algorithm, b) it can easily handle the appearance of new labels by training a new corresponding binary classifier, c) it can be easily parallelized to achieve a constant time complexity with respect to the number of labels[30,31].

This method is basically binary classification of labels. So it transforms original multi label dataset into $|L|$ single label dataset. It builds binary classifier for each label. For the classification of new instance, BR gives union of the labels that are positively predicted by $|L|$ classifier[32,33].

A relevant advantage of the BR approach is its low computational complexity compared with other Multi label methods. For a constant number of examples, BR scales linearly with size q of the label set L . Considering that the complexity of the base-classifiers is bound to $O(C)$, the complexity of BR is $qxO(C)$. Thus, the BR approach is quite appropriate for not very large q .

Merits of Binary Relevance method:

| Stream | n | p | n | n | p | p | n | n | n | n | n | n | n | p | p | p | p | n | n |
|---------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MW approach | | | | | * | * | | | | | | | * | * | * | * | * | * | * |
| MW(Buffer) approach | | | | | * | * | | | | | | | | * | * | * | * | * | * |

Fig. 3: Examples selected by a typical MW approach and by MW (Buffer) approach. The rightmost example is the most recent. A star indicates that the corresponding example appears inside the window. The size of the window is 10 in both the approach and the ratio of positive to negative examples in MW is 7/5. In MW (Buffer) is 7/2.

B. Space-Efficient Implementation of Multiple-Window with Buffer:

In the following, we describe a space-efficient implementation of the proposed multiple windows with buffer scheme. In particular, we discuss the update of window when a new example arrives. The pseudocode of the update method is listed in Algorithm 1. Table 1 summarizes the notation used in the pseudocode.

| Notation | Description |
|---|----------------------------|
| x_i | The i th stream instance |
| $Y_i = \{l_1, \dots, l_{y_i}\}$ | The label set of x_i |
| $L = \{l_1, \dots, l_L\}$ | Set of observed label |
| $B = \{(x_1, Y_1), \dots, (x_n, Y_n)\}$ | Shared buffer example |
| $Q_p = \{Q_1, \dots, Q_n\}$ | Positive Windows |
| $Q_n = \{Q_1, \dots, Q_n\}$ | Negative Windows |
| np | Size of positive window |

Table 1: Description of notation used in the method

Algorithm 1 is invoked for each arriving labeled instance, in order to update the positive and negative windows of each label. Each window is implemented as a queue. When the UpdateWindow function is called (lines 8, 10, 14), we insert the current instance in the queue and push the oldest instance out of the queue if it is full. The positive and negative queues store only references to the original instances, which are stored only once in the shared buffer B . Every time an instance is removed from a queue, the algorithm updates a counter which holds the number of

- It is Simple binary classification calculation and relatively fast.
- The independent modelling of BR allows handling the expected differences in frequencies and drift rates of the labels
- Can be coupled with any other classifier
- It can be parallelized to achieve constant time complexity with respect to the number of labels.

Demerits of Binary Relevance method:

- It does not consider label correlation ship between each correlated labels.

Another extension of BR method is BR+ method.

In this method, it should be observed that BR+ does not make any attempt to discover label dependency in advance. The main idea of BR+ is to increment the feature space of the binary classifiers to let them discover by themselves existing label dependency. In the training phase, BR+ works in a similar manner to BR, i.e., q binary classifiers are generated, one for each label $y_j \in L$. However, there is a difference related to the q binary datasets used to generate the binary classifiers. In BR+, the feature space of these datasets is incremented with $q - 1$ features, which correspond to the other labels in the multi-label dataset.

queues in which the instance is still present. If the counter becomes 0, the instance is also deleted from the shared buffer. Notice that when a labeled instance contains a label which appears for the first time, then the set of observed labels L is updated and a new positive and a new negative queue are created for this label (lines 12-13). Thus the algorithm automatically handles unseen labels. The size of the shared buffer $|B|$ determines the space complexity of the method and depends on the following factors :(1) the size of the positive windows, (2) the number of observed labels.

1) Algorithm 1: Updatemodel (X_i, Y_i)

- Input: Q_p, Q_n, L
- Output: The Updated Model
- 1) $B \leftarrow B \cup \{x_i, Y_i\}$
- 2) if $Q_p > T.V.$ then
- 3) $L \leftarrow B$
- 4) elseif $Q_p = Q_n$ then
- 5) $L \leftarrow B$
- 6) foreach $l_j \in L \cup Y$ do
- 7) if $l_j \in L \cap Y$ then
- 8) $Q_p \leftarrow \text{UpdateWindow}(x_i, Q_p)$
- 9) elseif $l_j \in L \setminus Y$ then
- 10) $Q_n \leftarrow \text{UpdateWindow}(x_i, Q_n)$
- 11) else
- 12) $L \leftarrow L \cup \{l_j\}, Q_p \leftarrow \text{null}, Q_n \leftarrow \text{null}$
- 13) $Q_p \leftarrow Q_p \cup Q_{pl}, Q_n \leftarrow Q_n \cup Q_{nl}$
- 14) $Q_{pl} \leftarrow \text{UpdateWindow}(x_i, Q_{pl})$

C. An Efficient Naïve-Bayes Implementation:

We chose naive-bayes to instantiate the proposed multiple windows approach for the following reasons:

- Naïve Bayesian classifiers assume that there are no dependencies amongst attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, hence is called "naive". This classifier is also called idiot Bayes, simple Bayes, or independent Bayes.
- It uses a very intuitive technique. Bayes classifiers, unlike neural networks, do not have several free parameters that must be set. This greatly simplifies the design process.
- Since the classifier returns probabilities, it is simpler to apply these results to a wide variety of tasks than if an arbitrary scale was used.
- It does not require large amounts of data before learning can begin.
- Naive Bayes classifiers are computationally fast when making decisions[42,43,44].

IV. EMPIRICAL RESULTS

A. Datasets:

We experimented on three large real-world textual multi-label datasets. Table 2 presents the main statistical properties of these datasets.

The tmc2007[52] dataset comes from a competition organized by the text mining workshop of the 7th SIAM international conference on data mining. The original data comprised 28596 aviation safety reports in free text form, annotated with one or more out of 22 problem types that appear during flights. The reuters (rcv1v2)[53,54] data set is a well known benchmark for text classification. It contains 804414 news articles over 365 days assigned to one or more out of 103 topics.

Text representation in all datasets follows the bag-of-words model. Boolean vectors are used in tmc2007, while tf-idf vectors are used in the case of rcv1v2. We applied feature selection to tmc2007 and rcv1v2 to select the top 500 features according to the χ^2 max criterion as described in. Note that both the calculation of the tf-idf vectors as well as the feature selection process are based on the complete dataset, hence they would not be feasible under a real data-stream environment. A dynamic feature space method should be used instead, but this is outside of the focus of this paper. Note also that tmc2007 is in fact static datasets with no specific instance ordering. We treat these datasets as streams and process them in their default order. On the other hand all instances in rcv1v2 are time ordered and are considered in this order.

| Name | D | X | L |
|---------|--------|------|-----|
| tmc2007 | 28596 | 500b | 22 |
| rcv1v2 | 804414 | 500n | 103 |

Table 2: Multi-Label Data Sets and their statistics. |D|: number of instances, |X|: number of attributes (b: binary/n: numeric), |L|: number of labels.

B. Baselines and Settings for All Algorithm:

We compare the performance of our method, denoted Multiple Window(Buffer) with Multiple Window approach

which contain two fixed size moving window one for positive and one for negative examples.

The MW (Buffer) approach were used a Naïve Bayes classifier while MW approach were used a kNN classifier in order to compare them with the proposed approach on the same basis.

C. Evaluation Measures & Methodology:

To evaluate the effectiveness of the methods we use the train-then-test (or prequential) evaluation methodology, where each example is first classified using the current classification model and it is then used to update the model. This way the classifier is tested against all stream examples before seeing them.

We use two measures to evaluate the effectiveness for a single label. The first one is F_1 , the harmonic mean of recall and precision, while the second one is the area under an ROC curve (AUC). AUC is appropriate for threshold independent evaluation, since it is calculated based on the confidence scores given by a classifier. We use macro-averaging to calculate a single measure across all labels, because it gives equal weight to each label in contrast to micro-averaging, which is dominated by high frequency labels.

To show the evolution of the models we calculate and report their performance every $|D|/20$ instances (approximately), on the previous $|D|/20$ instances. In addition, we report the macro-averaged F_1 for all stream instances to get an impression of the overall performance. Unfortunately, the calculation of macro-averaged AUC over all stream instances was infeasible, since it requires storing all the confidence scores from the beginning until the end of the stream, consuming all the available memory.

D. Results:

| Accuracy Measured | MW(Buffer) | | MW | |
|--------------------|------------|------|------|------|
| | no | th | no | th |
| F-measure(tmc2007) | 0.88 | 0.84 | 0.41 | 0.51 |
| ROC-Area(tmc2007) | 0.84 | 0.63 | 0.39 | 0.38 |
| F-measure(rcv1v2) | 0.64 | 0.88 | 0.23 | 0.40 |
| ROC-Area(rcv1v2) | 0.67 | 0.92 | 0.24 | 0.42 |

Table 3: Result on macro F_1 (F-measure)

Table 3 reports the results on the threshold dependent macro-averaged F_1 measure for all datasets. For each method, the two columns show the performance with and without thresholding.

We first notice that all two methods have substantial gains in macro-averaged F_1 when the thresholding technique is utilized. This shows that the proposed thresholding technique works quite well and at the same time stresses the importance of using a proper thresholding strategy for multi-label classification of data streams.

Next, we note that MW (Buffer) is better than MW in macro-averaged F_1 in all datasets, both with and without thresholding. This shows again that the multiple windows with buffer approach give the best result in all situations[26].

Figures 4, 5, 6 and 7 present the F-measure, ROC-Area, Kappa-Statistics and Mean-Absolute-Error of tmc2007 and rcv1v2 respectively. The proposed multiple windows with buffer approach consistently outperforms multiple windows approach on all datasets in this threshold-independent evaluation. The boost in performance is more apparent in rcv1v2.

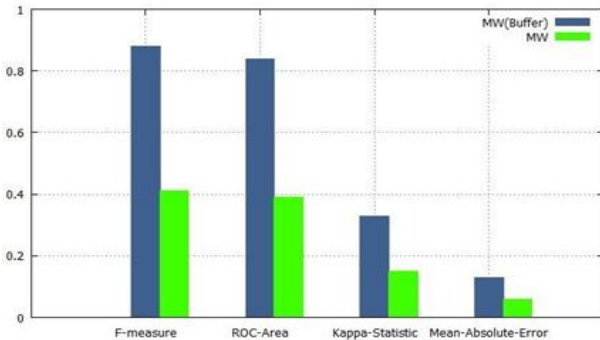


Fig. 4: Macro AUC (F-measure) result on tmc2007

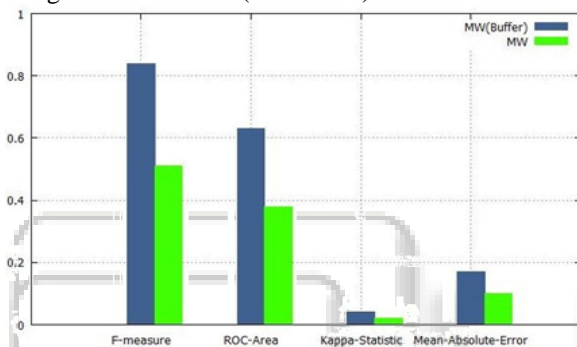


Fig. 5: Macro AUC (F-measure) result on tmc2007

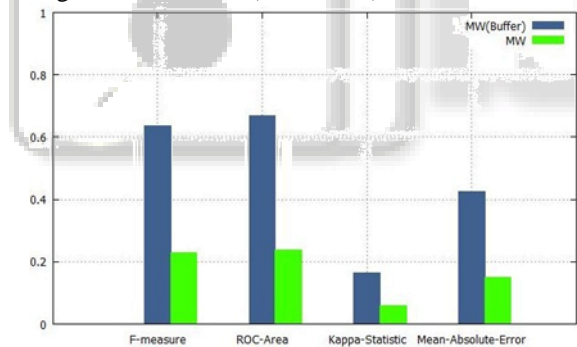


Fig. 6: Macro AUC (F-measure) result on rcv1v2

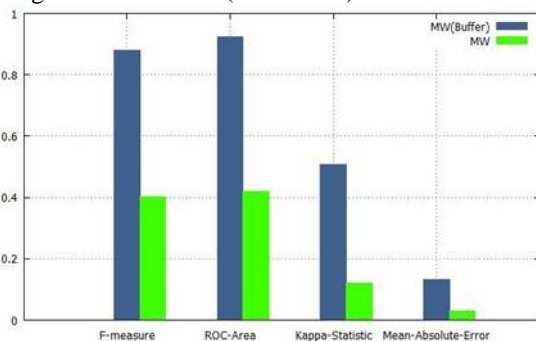


Fig. 7: Macro AUC (F-measure) result on rcv1v2

V. CONCLUSIONS AND FUTURE WORK

We presented a novel method for multi-label stream classification which adopts a multiple windows approach with buffer MW (Buffer) to deal with concept drift and

skewness in the distribution of positive and negative examples of each label. In general we get more positive examples as compared to Negative examples in datasets with this approach. Our method, being independent of the base classifier, offers a general framework for dealing with evolving multi-label streams. Space and time efficient implementations of this method were discussed.

The empirical evaluation showed that 1) the theoretical advantage of our learning method is verified in practice by substantial gains in positive to negative examples in datasets using Buffer approach.

In the future we plan to 1) give our method the ability to model label correlations by utilizing methods, 2) dynamically adjust the size of the positive window of each label using a drift detection method, 3) employ a mechanism to efficiently deal with label set expansion, 4) experiment with different binary base classifiers and tree classifiers and 5) evaluate our method in synthetic datasets modelling various concept drift patterns and imbalance degrees.

REFERENCES

- [1] Ethem Alpaydin. "Introduction to Machine Learning". The MIT Press, 2nd edition, 2010.
- [2] Tom M. Mitchell. "Machine Learning". McGraw-Hill Education, 1997.
- [3] Svitlana Volkova. "Data Stream Mining: A Review of Learning Methods and Frameworks".
- [4] Domingos, P. and Hulten, G. (2000). "Mining high-speed data streams", in Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 71-80.
- [5] Elena Ikonomovska, Suzana Loskovska. "A Survey of stream data mining". In Eighth National conference with international participation – ETAI 2007.
- [6] Lichtenwalter, R. N. and Chawla, N. V. (2010). "Adaptive methods for classification in arbitrarily imbalanced and drifting data streams", in Proceedings of the 13th Pacific-Asia international conference on Knowledge discovery and data mining: new frontiers in applied data mining, PAKDD'09, Springer-Verlag, Berlin, Heidelberg, pp. 53-75.
- [7] Haixun Wang, Wei Fan, Philip S. Yu, Jiawei Han. "Mining Concept-Drifting Data Streams using Ensemble Classifiers", in Proceedings of the 1st Asian Conference on Machine Learning, pp. 308-321.
- [8] Wang, H., Fan, W., Yu, P. S. and Han, J. (2003). "Mining concept-drifting data streams using ensemble classifiers", in Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '03, ACM, New York, NY, USA, pp. 226-235.
- [9] Albert Bifet, Ricard Gavaldà. "Mining Adaptively Frequent Closed Unlabeled Rooted Trees in Data Streams". In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '04, ACM, New York, NY, USA, pp. 503-508.
- [10] Katakis, I., Tsoumakas, G. and Vlahavas, I. (2006). "Dynamic feature space and incremental feature

- selection for the classification of textual data streams”, in ECML/PKDD-2006 International Workshop on Knowledge Discovery from Data Streams, Springer Verlag, p. 107.
- [11] Cramer, K. and Singer, Y. (2003). “A family of additive online algorithms for category ranking”, *J. Mach. Learn. Res.* 3, 1025-1058.
- [12] Gama, J., Sebastiao, R. and Rodrigues, P. (2009). “Issues in evaluation of stream learning algorithms”, in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 329-338.
- [13] Hulten, G., Spencer, L. and Domingos, P. (2001). “Mining time-changing data streams”, in Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '01, ACM, New York, NY, USA, pp. 97-106.
- [14] Purvi Prajapati, Amit Thakkar, Amit Ganatra. ”A Survey and Current Research Challenges in Multi-Label Classification Methods”. In *International Journal of Soft Computing and Engineering (IJSCE)* ISSN: 2231-2307, Volume-2, Issue-1, March 2012. Pp. 248-252.
- [15] Hullermeier, E., Furnkranz, J., Cheng, W. and Brinker, K. (2008). “Label ranking by learning pairwise preferences”, *Artif. Intell.* 172, 1897-1916.
- [16] Jesse Read Albert Bifet Geoff Holmes Bernhard Pfahringer. ”Streaming Multi-label Classification”. *JMLR In Workshop and Conference Proceedings* 17 (2011) 19-25 2nd Workshop on Applications of Pattern Analysis.
- [17] S. Godbole and S. Sarawagi. ”Discriminative methods for multi-labeled classification”. In Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining, pages 22–30, 2004.
- [18] Outline: Multi Label Classification <http://www.tsc.uc3m.es/~jesse/talks/mend.pdf>.
- [19] Grigorios Tsoumakas, Ioannis Katakis. ”Multi-Label Classification: An Overview”. *International Journal of Data Warehousing and Mining*, David Taniar (Ed.), Idea Group Publishing, 3(3), pp. 1-13, 2007.
- [20] Ghamrawi, N. and McCallum, A. (2005). “Collective multi-label classification”, in Proceedings of the 14th ACM international conference on Information and knowledge management, CIKM '05, ACM, New York, NY, USA, pp. 195-200.
- [21] Andreas P. Streich and Joachim M. Buhmann. “Classification of Multi-labeled Data: A Generative Approach”, W. Daelemans et al. (Eds.): ECML PKDD 2008, Part II, LNAI 5212 Springer-Verlag Berlin Heidelberg 2008, pp. 390–405, 2008.
- [22] Ueda, N. and Saito, K. (2003). “Parametric mixture models for multi-labeled text”, in *Advances in Neural Information Processing Systems 15* MIT Press, pp. 721-728.
- [23] Zhu, S., Ji, X., Xu, W. and Gong, Y. (2005). “Multi-labelled classification using maximum entropy method”, in Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '05, ACM, New York, NY, USA, pp. 274-281.
- [24] Lewis, D. D. (1995). “Evaluating and optimizing autonomous text classification systems”, in Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '95, ACM, New York, NY, USA, pp. 246-254.
- [25] Yiming Yang. “A Study on Thresholding Strategies for Text Categorization”, SIGIR'01, September 9-12, 2001, New Orleans, Louisiana, USA. Copyright 2001 ACM 1-58113-331-6/01/0009.
- [26] Eleftheros S xioufis, Myra Spiliopoulou, Grigorios tsoumakas. “Dealing with concept drift and class imbalance in multi-label stream classification”. In proceedings of the twenty-second international joint conference on artificial intelligence, February 2011.
- [27] Gao, J., Fan, W., Han, J. and Yu, P. (2007). “A general framework for mining concept-drifting data streams with skewed distributions”, in Proceedings of the 7th SIAM International Conference on Data Mining (SDM'07), pp. 3-14.
- [28] Everton Alvares Cherman, Maria Carolina Monard, Jean Metz. ”Multi-label Problem Transformation Methods a Case Study”. In *CLEI ELECTRONIC JOURNAL*, VOLUME 14, NUMBER 1, PAPER 4, APRIL 2011.
- [29] Clare, A. and King, R. (2001). “Knowledge discovery in multi-label phenotype data”. In Proceedings of the 5th European Conference on Principles of Knowledge Discovery in Databases, pp. 42-53.
- [30] Erica Akemi Tanaka and Jose Augusto Baranauskas. ”An Adaptation of Binary Relevance for Multi-Label Classification applied to Functional Genomics”. In Proceedings of the XXXII Congress of the Brazilian Computer Society, XII Workshop on Medical Informatics, Curitiba, PR, Brazil, July 16-19, 2012, ISSN 2175-2761”.
- [31] Tsoumakas, G., Dimou, A., Mezaris, V., Kompatsiaris, I. and Vlahavas, I. (2009). “Correlation-based pruning of stacked binary relevance models for multi-label learning”, in Proc. ECML/PKDD 2009 Workshop on Learning from Multi-Label Data, MLD'09.
- [32] Dawid, A. P. (1984). “Present position and potential developments: Some personal views: Statistical theory: The prequential approach”, *Journal of the Royal Statistical Society. Series A (General)* 147(2), pp. 278-292.
- [33] Klaus Brinker and Johannes Furnkranz and Eyke Hullermeier. ”A Unified Model for Multilabel Classification and Ranking”. In Proceedings of the 2006 conference on ECAI.
- [34] Jesse Read, Bernhard Pfahringer, Geoff Holmes, Eibe Frank. ”Classifier Chains for Multi-label Classification”.
- [35] B. Reshma Yusuf, Dr. P. Chenna Reddy. ”Mining Data Streams using Option Trees”. *Computer*

- Engineering and Intelligent Systems ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online) Vol 3, No.7, 2012.
- [36] Xiangnan Kong, Philip S. Yu. "An Ensemble-based Approach to Fast Classification of Multi-label Data Streams".
- [37] Mohammad M. Masud, Clay Woolam, Jing Gao, Latifur Khan, Jiawei Han, Kevin W. Hamlen, Nikunj C. Oza. "Facing the reality of data stream classification: coping with scarcity of labeled data". © Springer-Verlag London Limited 2011. Received: 6 May 2009 / Revised: 26 April 2011 / Accepted: 22 October 2011.
- [38] Jesse Read, Albert Bifet, Geoff Holmes, Bernhard Pfahringer. "Efficient Multi-label Classification for Evolving Data Streams". In Working Paper Series ISSN 1177-777X.
- [39] Mccallum, A. (1999). "Multi-label text classification with a mixture model trained by EM", in Proceedings of the AAAI'99 Workshop on Text Learning.
- [40] Peng Wang, Peng Zhang, Li Guo. "Mining Multi-label Data Streams Using Ensemble-based Active Learning", in Proceedings of the 1st Asian Conference on Machine Learning.
- [41] Min-Ling Zhang and Zhi-Hua Zhou, "Multi-Label Learning by Instance Differentiation", In Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence AAAI Press, pages 669-674, Vancouver - Canada, 2007.
- [42] Cheng, W. and Hullermeier, E. (2009). "Combining instance-based learning and logistic regression for multilabel classification". *Mach. Learn.* 76, 211-225.
- [43] Read, J., Pfahringer B., Holmes G. Dept. of Computer Sci., Univ. of Waikato, Hamilton. "A Pruned Problem Transformation Method for Multi-Label Classification". In Data Mining, 2008. ICDM '08. Eighth IEEE International Conference, pages: 995 – 1000, 15-19 Dec. 2008.
- [44] Jesse Read, Bernhard Pfahringer, Geoff Holmes. "Generating Synthetic Multi-label Data Streams".
- [45] Ueno, K., Xi, X., Keogh, E. and Lee, D.-J. (2006). "Anytime classification using the nearest neighbour algorithm with applications to stream mining", IEEE International Conference on Data Mining, pp. 623-632.
- [46] Schapire, R. E. and Singer, Y. (2000). "Boostexter: A boosting-based system for text categorization", *Machine Learning* 39, 135-168.
- [47] Grigorios T, Eleftherios Spyromitros-Xioufis, Jozef Vilcek, Ioannis Vlahavas. "MULAN: A Java Library for Multi-Label Learning". In *Journal of Machine Learning Research* 12 (2011) 2411-2414 Submitted 8/09; Revised 7/10; Published 7/11.
- [48] Learning from Multi Label Data <http://mlkd.csd.auth.gr/multilabel.html>.
- [49] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I. H. (2009). "The weka data mining software: an update", *SIGKDD Explor. Newsl.* 11, 10-18.
- [50] Bifet, A., Holmes, G., Kirkby, R. and Pfahringer, B. (2010). "Moa: Massive online analysis", *J. Mach. Learn. Res.* 11, 1601-1604.
- [51] Albert Bifet, Geoff Holmes, Bernhard Pfahringer, Philipp Kranen, Hardy Kremer, Timm Jansen and Thomas Seidl. "MOA: Massive Online Analysis, a Framework for Stream Classification and Clustering."
- [52] A. Srivastav, B. Zane-Ulman. "Discovering recurring anomalies in text reports regarding complex space systems". In *Proceedings of IEEE Aerospace Conference*, pages 3853 –3862, 2005.
- [53] J. Read, B. Pfahringer, G. Holmes, and E. Frank. "Classifier chains for multi-label classification". In *Proceedings of ECML PKDD '09*, pages 254–269, 2009.
- [54] Lewis, D., Yang, Y., Rose, T. and Li, F. (2004). "Rcv1: A new benchmark collection for text categorization research", *Journal of Machine Learning Research* 5, 361-397.
- [55] D.D. Lewis, Y. Yang, T.G. Rose and F. Li. Rcv1. "A new benchmark collection for text categorization research". In *Journal of Machine Learning Research*, 5:361–397, December 2004.
- [56] Jin, R. and Agrawal, G. (2003). "Efficient decision tree construction on streaming data", in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '03*, ACM, New York, NY, USA, pp. 571-576.