Recurring Concept Detection for Spam Filtering

Miguel Ángel Abad *, João Bártolo Gomes †, Ernestina Menasalvas *
*Universidad Politecnica de Madrid, Spain
Email: miguel.abad.arranz@alumnos.upm.es, emenasalvas@fi.upm.es
†Institute for Infocomm Research (I2R), A*STAR, Singapore
Email:bartologjp@i2r.a-star.edu.sg

***Abstract*—In this work we dig into the problem of recurring concept drifts, proposing a framework to manage them. Its implementation and evaluation phases have been oriented to solve the spam detection problem, taking into account that it is a real-world situation where concepts (spam patterns) may reappear. The possibility of detecting recurring drifts allows to reuse previously learnt models, enhancing the overall learning process specifically in terms of accuracy and efficiency. Consequently, in this paper we propose the Meta-Model Drift Detector (MM-DD). The proposed system is able to deal with the underlying context that results from the drifts detected throughout the data stream learning process. In order to do so, a meta-model is trained in parallel to the learning process. While the learning process of the base classifier is feeding the meta-model with all the context information when a drift occurs, the later is able to predict in the near future recurrent situations. Therefore, when a drift is detected the meta-model checks if the context information is equal to any of the previously managed by the learning process and provides the most suitable stored model to deal with the concept. Our experimental results support the value of the proposed MM-DD in terms of accuracy when compared with existing approaches.**

I. Introduction

Nowadays we live in a world where many technologies continuously generate more and faster flows of information. Although these flows, known as data streams, are part of our daily lives, it is still difficult to deal with them in an efficient way. In the scope of data mining it is a real challenge to deal with such kind of information as the data cannot be stored in memory for analysis due to is unbounded nature. Examples of data streams are banking transactions, credit card operations, sensor networks, network communications, collaborative platforms, news feeds or communications via e-mail [1], [2].

Traditional data stream classification [3], [1] aims to learn a classification model from a stream of training records in order to use it later to predict the class of unlabelled records with high accuracy. Most of these kinds of classification models lack an efficient adaptation to the environment where they are implemented which, in most cases, is constantly changing. For this reason, coping with the improvement and adaptation of classification algorithms on data streams is still a great challenge.

In order to solve such problems, in the scope of data streams classification algorithms, concept drift techniques have been extensively applied. These techniques allow to handle changes in the distribution of instances during the learning process. In this way, concept drift techniques allow the classification models to adapt their learning process to the different distribution changes when they are appearing [4], [5], [6].

In fact most of the real-world scenarios based on mining data streams are characterised by the appearance of concept drifts. Moreover, although concept drift is usually defined as the unforeseeable concept change of the class variable in a prediction process, there are some situations where such unforeseeable changes can be detected. Such is the case of recurring concept drifts, where a method could be implemented to learn from past drifts and detect if any of them is taking place again in a specific time. However most of the classification models adapt to recurrent situations by means of training a new model from scratch. That is, they do not cope with recurrent drifts because they do not even detect that recurrence.

Taking into account that concept changes are generally the result of context changes, the context information related to a drift could be used to better understand recurring concept changes. However, only a small number of techniques explore context information when dealing with recurring concept changes [7], [8], [9].

If the learning system could remember recurrent concepts, both the performance and accuracy of the classification algorithm could be enhanced. This method should cope with the ability of detecting which is the most suitable previously learnt model for a certain context. In order to minimise the impact on the current classification model involved in the learning process, a possible solution would be by means of a meta-model trained in parallel. In this way, the meta-model aims to learn from the changes in the classification task providing an early detection mechanism of recurrent drifts. In fact, we could apply a previously trained model thus not needing a new training process.

In order to deal with recurrent concept drift on data stream environments two main approaches arise:

- To adapt the algorithm itself to include concept drift capacities.
- To develop a framework capable of being used on different environments, and being transparent to the actual classification algorithm in use.

There are several pros and cons for each choice, but the framework solution appears to be more versatile because it can use several different data stream classification algorithms in a transparent way. In the case of adapting the algorithms, we would need to re-implement every algorithm subject of being used in a specific environment.

Taking the framework option, in this paper, we propose MM-DD as a novel data-stream learning system framework to aid in the process of recurrent concept drift management. It is based on the establishment of a meta-model trained to predict when the drift will occur as well as the most suitable model to be reused if necessary.

The main contribution of the paper is the development of a framework to manage recurrent concept drifts, being transparent of the specific classification and drift detection algorithms in use. In order to test the benefits and utility of dealing with recurrent concepts, an instantiation of this framework has been evaluated against three datasets. Two of them are related to spam detection and email filtering and the other being a synthetic dataset.

The MM-DD framework proposed in this paper improves the classification system capabilities in the following way:

- Increasing the knowledge of the context information available behind each concept drift.
- Training a classification meta-model from the context information, that it is able to predict recurrent situations.
- Managing the recurrent concepts, reusing previously seen models if needed.

The the context of spam detection mechanisms, avoiding the need of re-learning previously seen concepts can be beneficial. For example, a concept would be a specific pattern related to a group of spam messages. In this way, when the spam filtering has learnt a recurring pattern, it could re-use it in the MM-DD framework, saving training instances in the learning process.

The rest of the paper is organised as follows. In Section II, we summarise related work on concept drift and context-aware approaches in recurring concepts. This is followed in Section III by the preliminaries of the approach where the motivation, challenges and problem definition are stated. Furthermore, in Section IV, we propose MM-DD framework as a solution to work in recurring concept drift environments, with a detailed description of its components and the algorithm used. Section V introduces the experimental setup and the datasets used to evaluate MM-DD framework. This is followed by the a detailed discussion of the results of our evaluation. Finally, in Section VI, our conclusions and possible topics for future research are presented.

II. Related Work

There have been several techniques developed to achieve the challenge that arises when dealing with concept drift [4], [6], that can be organised into three main groups, namely instance selection, instance weighting and ensemble methods [4]. In this way, different new algorithms have appeared to detect concept drift [10], [11], [12], [5], [3], [6], but it is not so common for them to detect and manage recurrent changes.

Other approaches that by means of combining drift detection [5] and storage of learnt models allow to reuse them in case of concept recurrence [13], [14], [15]. An improvement is made by these proposals, as long as they avoid the extra effort needed when re-learning a previously managed concept. Specifically, the method proposed by Yang et al. [14] is based on a proactive approach, which means reusing a concept from the concept history; in the case of the solution proposed in[15], an ensemble method is used. However, although there are proposals that detect recurrence, in some cases the models applied must still keep on learning, as occurs in [16]. That is to say that although the recurrence is detected, the process is not completely optimised to save costs on time and performance[17], [9].

The relationship that may exist between a concept and the context where it appears has already been stated as a problem to be solved in several real world domains [18], [7], [8], [19]. Turney [18] was among the first to introduce the problem of context in machine learning. In his work, a formal definition was presented in which the notions of primary, contextual and context-sensitive features were introduced.

In the approach proposed in [17], [9], [20] context-concept relationships are learnt from the concept history. A model from a previously learnt concept associated with a particular context is reused in situations of recurrence. Extending on this dependance between context and concept drifts, in this work we propose a framework that counts with a meta-model as the main pillar of its development. From the fact that neither of the former approaches explore the need to develop a mechanism to represent concept drifts nor the ability of learning from them, we propose an innovative way of dealing with recurrent drifts management.

III. Methods

The main goal of this section is to provide all the basis needed to understand the proposed MM-DD framework. This section provides a comprehensive overview of the basic concepts and challenges related to the solution proposed. The main premise from now on is that we are assuming that the data streams used as input of the classification learning system are already preprocessed. In this way, transformation of the raw data into features used in the classification learning process is out of the scope of this work.

A. Learning with Concept Drift

Let X be the space of attributes with its possible values; Y the set of possible discrete class values. Let D be the data stream of training records arriving sequentially $X_i = (\vec{x_i}\ , y_i)$ with $x_i \in X$ (feature space) and $y_i \in Y$, where $\vec{x_i}$ is a vector of the attribute values and y_i is the (discrete) class label for the i^{th} record in the stream. In order to train a base learner based on a classification model m incrementally, these records are processed by m with the goal of predicting the class label of a new record $\vec{x} \in X$, so that $m(\vec{x}) = y \in Y$.

In this field, we consider that a stable concept has been learnt when the records used during a given period k are independently and identically distributed according to a probability distribution $P_k(x, y)$. In those situations where concept change, $P_k(x, y) \neq P_{k+1}(x, y)$.

B. Recurrent Concept Drift

After some time, the classification model may have been replaced by a new model $m + 1$ because of a concept change. In this case, when a set of training records $X_{i+k} = (\vec{x_{i+k}}\ ,$

y_{i+k}) with $x_{i+k} \in X$ (feature space) and $y_{i+k} \in Y$ arrives after a period k, it may be the case that the just arrived training set is equal to a set of records previously managed by the classification model m. In this case, the model m could be used again to deal with the new training set, avoiding the $m+1$ model to learn a concept that m has proven to manage well.

C. Multiple-instance classification

In order to predict recurrent drifts we have to make use of the context information associated to each drift. In this way, we need to count with a system able to link all the instances that appear during a drift with the new model used to deal with the change. With a system like that, we would be able to check if a set of instances are equal to a previously seen set. In those case, we could re-use the model saving training instance in the process. In order to solve the problem, the multiple-instance (MI) learning model is a good approach. In the MI learning model, each training example is composed by a set (bag) of instances along with the class label associated to that bag. The multi-instance learning model was formalised in [21].

IV. Solution Proposed

In order to deal with the recurrent concepts, we propose a framework to be used in the case of classification learning processes. This MM-DD framework allows a classification system to better adapt to recurrent situations, by predicting not just when a drift is going to appear, but also if that drift is a recurrent one. That prediction is made thanks to the disposal of a meta-model that trained on the past concept drifts can determine if another one is going to happen using a bunch of stream records at a particular time. But thanks to such a trained meta-model, the framework allows also to determine what is the most suitable previously seen model to be used on each particular case. These characteristics make MM-DD an effective framework to early detect drift and to obtain in a real scenario the best fitted model for a given context.

The innovative characteristics of the proposed MM-DD framework rely on two main elements presented in figure 1:

- A meta-model based on an extra classification algorithm trained from the context information associated to each specific concept drift. A meta-model as the one proposed allows to predict both when the drift will happen and the most suitable concept for each situation if it is recurrent.
- A repository to store all the previously seen and trained models related to the concepts involved in the drifts detected during the classification process.

The global functioning of the framework proposed can be seen as a three-layer system:

1) A basic layer where an incremental learning algorithm is able to represent the underlying concept by means of a classification model. In this layer the base learner processes the incoming records from data streams by means of an incremental learning algorithm to generate a decision model representing the underlying concept. The model will be used to classify unlabelled records.

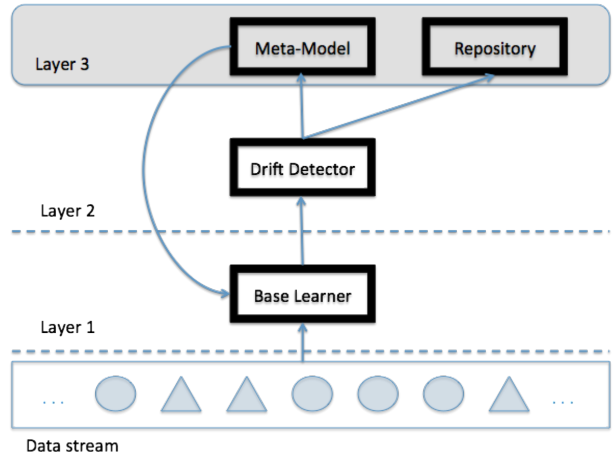

Fig. 1. MM-DD Components

2) A second layer where the detection of concept drift is implemented. In this case a drift detection method should be continuously monitoring the error-rate of the learning algorithm [5]. When the error-rate goes beyond some predefined levels, the drift detection method should signal a *drift*.
3) An extended layer in which detection and adaptation to recurrent concept changes takes place, by means of a meta-model data-mining process. In parallel to the functioning of the aforementioned layers, a meta-model is trained from the context information provided by the drift detection method.

Throughout the life cycle of the system, three different cases may be used to adapt to changes in the underlying concept, depending on the availability or not of a trained meta-model:

- The concept similarity method detects that the underlying concept is new, and the base learner has to learn it by processing the current incoming labelled records in an incremental way.
- The underlying concept is recurrent and a previous model is applied.
- The meta-model is able to predict the drift, and if it refers to a recurrent concept, it states the best model to be used from the repository.

In Algorithm 1 we can see the learning process of MM-DD. When a drift is detected by the drift detector mechanism in use, the meta-model is asked to look for a previously seen model stored in PM that could deal with the current underlying distribution associated to the current training set DS. If the meta-model finds such a suitable previously used model, the learning process makes use of it again as the current classifier.

V. Experiments

In order to test the utility and benefits of the proposed MM-DD framework a test-bed implementation has been developed for this work, based on the MOA [22] environment. The base learner classifier used is the Naive Bayes, and the drift detection method used is the one presented in[5], implemented in MOA through the *SingleClassifierDrift* class. The meta-model used in this experimentation phase has been implemented through the MIEMDD [23] algorithm.

Experiments to test the feasibility of the MM-DD learning framework have been developed in terms of:

- Accuracy. This parameter measures the percentage of the overall correctly predicted records.
- Kappa statistic. This parameter measures the classifier agreement and it is considered to be a more robust measure than simple percentage agreement calculation since Kappa takes into account the agreement with a random classifier. It is usually used in imbalanced learning problems like the spam filtering, where it is common to get a higher rate of normal messages than spam messages.
- Memory consumption of the classification models involved in the learning process.

All those measures have been also assessed in comparison to the methods:

- AUE ensemble method presented in [24]. This method is the state-of-art in adaptive ensemble classification. It works by breaking the stream into fixed-size data chunks to sequentially learn classifiers. These classifiers are evaluated and their combination weights are updated based on the accuracy results.
- Naive Bayes classifier. This simple and efficient classifier has been used successfully in many text mining problems [20], [25], [26]. It assumes feature independence and in this experiment it has been used as base learning classifier.

In order to assess the utility of the framework proposed in this work, two real and one synthetic datasets are used. The former are the ones presented in[15] publicly available in http://mlkd.csd.auth.gr/concept_drift.html, while the later is based on the SEA dataset with four different concepts presented in[12].

A. Emailing list dataset

This dataset represents the problem of sudden concept drift and recurring contexts. Its content simulates a stream of email messages that are sequentially presented to a user who then labels them as interesting or junk according to her personal interests. The goal is therefore to train a model being able to automatically classify messages into interesting or junk as they arrive.

Algorithm 1 Learning Process

Require: Data stream DS, PreviousModels PM

repeat
  Get next record DS_i from DS;
  prediction = *currentClassifier.classify(*DS_i*)*;
  DriftDetection.update(prediction);
  if $DriftDetection.isDrift()$ **then**
    PM.store(currentClassifier);
    predictedModel = metamodel.getPrediction(PM*,*DS*)*;
  end if
  if $\neg predictedModel.isEmpty()$ **then**
    currentClassifier = predictedModel;
  end if
until END OF STREAM

TABLE I. DATASETS PRECISION

Dataset	AUE		Naive Bayes		MM-DD	
	Acc.	Kappa	Acc.	Kappa	Acc.	Kappa
Emailing List	55.19	1.55	52.74	8.64	**61.76**	**25.89**
Spam	88.9	53.43	**91.24**	**71.05**	89.22	61.59
SEA	86.92	70.97	83.78	64.74	**87.52**	**72.24**

The dataset is a stream of 1,500 examples and 913 attributes which are words that appeared at least 10 times in the corpus (boolean bag-of-words representation). The context information that indicates a concept drift is based on the topic interest of the simulated user.

As it can be seen in table I, the MM-DD implementation used during this experimentation phase provides the best results both on accuracy and kappa statistic. In Figure 2 we can be seen that during the learning process, MM-DD implements does not need to count with all the stream records. This is because of reusing previously learnt models avoid the need of training them while keeping a good accuracy.

Lastly, it is also important to note the positive impact that reusing previously seen models have on the memory consumption of the system. As it is represented in Figure 3, the implementation of the MM-DD framework is the one that fewer number of bytes consumed when using the different classification models.

B. Spam filtering dataset

This dataset represents the problem of gradual concept drift, and is based on the Spam Assassin Collection available in http://spamassassin.apache.org/ using the boolean bag-of-words approach and the adaptations presented in[15]. This dataset consists of 9,324 instances and 500 attributes.

In this case, although the implemented MM-DD framework does not get the best precision results, it provides good results comparing with the other two algorithms used in the experimentation, as it can be seen in table I. However, the

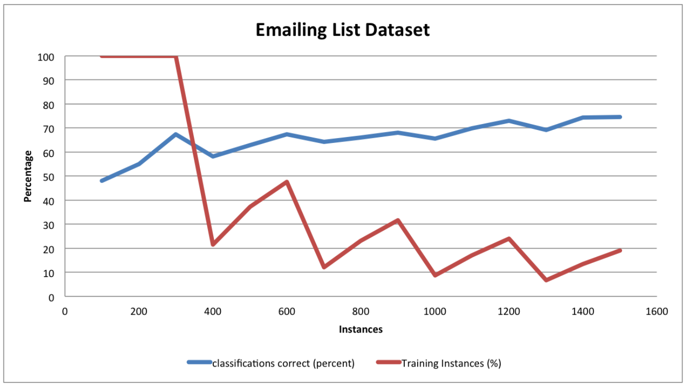

Fig. 2. Emailing List Dataset - Accuracy vs Training Instances

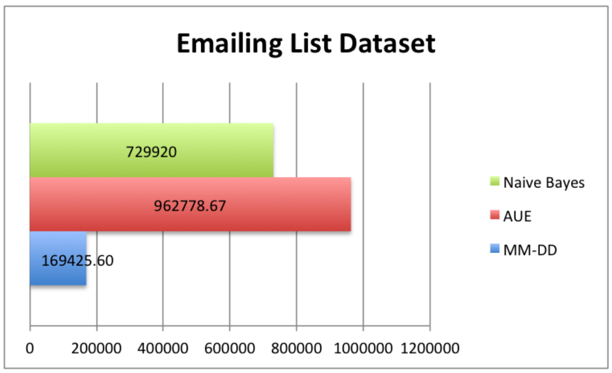

Fig. 3. Emailing List Model Size (bytes)

slightly lower results in accuracy may be originated by the previously trained models used during the learning process. As it can be seen in Figure 4, during the learning process the implementation of MM-DD makes a low use of training instances because of the aforementioned pre-trained models while keeping a high level of accuracy. Furthermore, in Figure 5 we can see that also the medium size of the models used is the lower among the different algorithms compared in this experimentation.

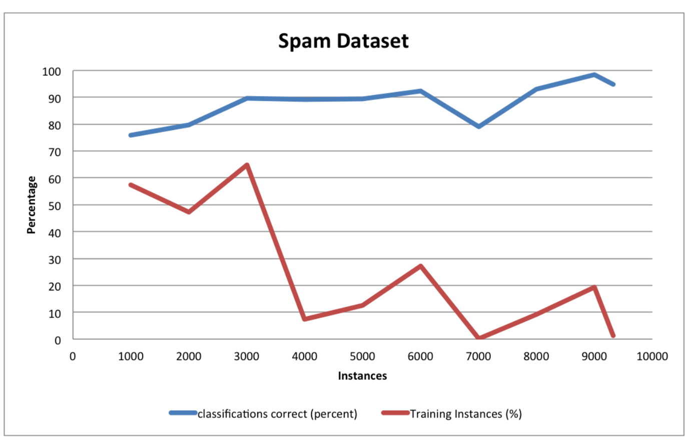

Fig. 4. Spam Dataset - Accuracy vs Training Instances

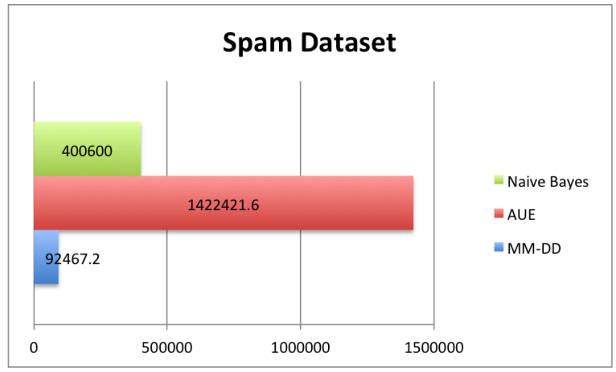

Fig. 5. Spam Model Size (bytes)

C. SEA dataset

This synthetic dataset is based on the work presented in[12]. It consists of 50,000 instances, each of them containing 4 attributes.

As it is represented in table I, in this case the implementation of MM-DD framework gets the best precision results, both in accuracy and kappa statistic. Also in this case some previously trained models are re-used, as it can be seen in the training streams needed during the process in Figure 6 . Again the implementation of MM-DD makes use of the lower medium size of models during all the learning process, as Figure 7 shows.

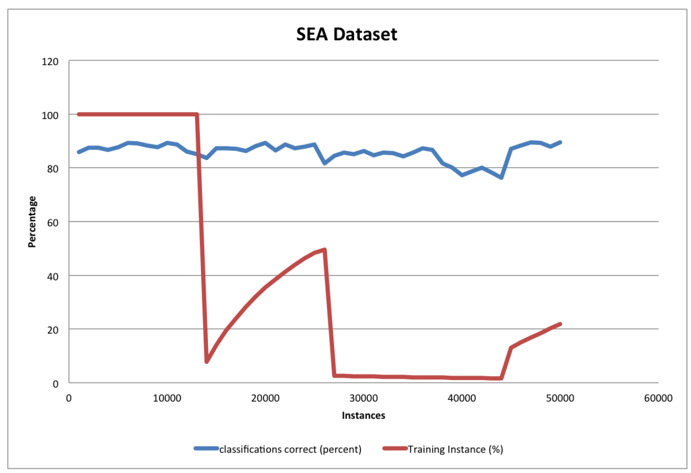

Fig. 6. SEA Dataset - Accuracy vs Training Instances

VI. Conclusions and Future Lines of Research

In this paper the MM-DD framework has been described as a mechanism to manage concept drift in recurring situations. The main contributions in MM-DD come from the availability of a parallel meta-model to deal with the prediction of recurrent concepts. This meta-model allows a classification system to re-use previously seen models that represent same concepts. This framework has proven to be valid in real-world scenarios as is shown in our experimental evaluation.

By using the MM-DD framework proposed in this paper, two main challenges are approached:

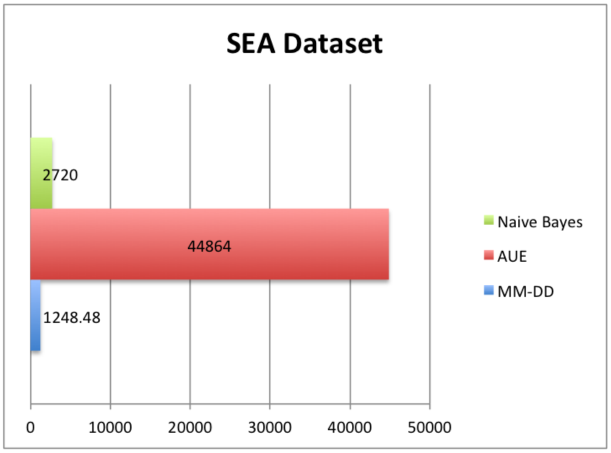

Fig. 7. SEA Model Size (bytes)

1) The possibility of implement a meta-model representing the underlying concept drifts detected throughout the lifetime of the base learner.
2) The development of MM-DD framework as a wrapper mechanism, that can be used with different base classification learners. Furthermore, this wrapper mechanism allows to parametrise the meta-model to meet the characteristics of the environment where it is being used.

An implementation of MM-DD framework has been tested on different synthetic and real datasets, and comparisons have been made with another similar context-aware algorithm as well as with the traditional Naive Bayes classifier. The main conclusions obtained from those experiments are that MM-DD:

- outperforms in most cases the precision results provided by the other evaluated state-of-art algorithms.
- makes a lower use of training instances during the learning process.
- provides the lower medium size of models used during all the learning process.

In contrast, the main drawback of MM-DD framework is the fact that in order to fit the necessities of a real-world scenario, the parameters of the meta-model have to be deeply assessed.

The proposed framework has been proved to be a useful way of dealing with spam detection filtering, due to the fact that a spam pattern is likely to reappear. A meta-model framework would also allow to know the hidden behaviour of the concept changes. Such knowledge would let the classification system to adapt to each situation in a more efficient way.

The future lines of research to analyse possible improvements of MM-DD framework are: a) the implementation of a function able to predict not just recurrent situations, but also similar ones; b) the establishment of a loss function to penalise bad predictions when reusing previously seen models, and; c) the evaluation of the actual computational cost of the meta-model, and its optimisation to real-world environments.

References

[1] M. M. Gaber, J. Gama, S. Krishnaswamy, J. B. Gomes, and F. Stahl, "Data stream mining in ubiquitous environments: state-of-the-art and current directions," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, no. 2, pp. 116–138, 2014.

[2] M. M. Gaber, J. B. Gomes, and F. Stahl, "Pocket data mining," 2014.

[3] M. Gaber, A. Zaslavsky, and S. Krishnaswamy, "A survey of classification methods in data streams," in *Data Streams - Models and Algorithms*, ser. Advances in Database Systems, A. K. Elmagarmid and C. C. Aggarwal, Eds. Springer, 2007, vol. 31, ch. 3, pp. 39–59.

[4] A. Tsymbal, "The problem of concept drift: definitions and related work," *Computer Science Department, Trinity College Dublin*, 2004.

[5] J. Gama, P. Medas, G. Castillo, and P. P. Rodrigues, "Learning with drift detection," in *Advances in Artificial Intelligence, Proceedings of SBIA 2004*, ser. Lecture Notes in Computer Science, A. L. C. Bazzan and S. Labidi, Eds., vol. 3171. Springer, 2004, pp. 286–295.

[6] I. Žliobaitė, "Learning under concept drift: an overview," *Technical Report. Faculty of Mathematics and Informatics, Vilnius University: Vilnius, Lithuania.*, 2010.

[7] G. Widmer, "Tracking context changes through meta-learning," *Machine Learning*, vol. 27, no. 3, pp. 259–286, 1997.

[8] M. Harries, C. Sammut, and K. Horn, "Extracting hidden context," *Machine Learning*, vol. 32, no. 2, pp. 101–126, 1998.

[9] J. B. Gomes, P. A. Sousa, and E. Menasalvas, "Tracking recurrent concepts using context," *Intelligent data analysis*, vol. 16, no. 5, pp. 803–825, 2012.

[10] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine learning*, vol. 23, no. 1, pp. 69–101, 1996.

[11] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM New York, NY, USA, 2001, pp. 97–106.

[12] W. N. Street and Y. Kim, "A streaming ensemble algorithm (SEA) for large-scale classification," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2001, pp. 377–382.

[13] J. Gama and P. Kosina, "Tracking Recurring Concepts with Meta-learners," in *Progress in Artificial Intelligence: 14th Portuguese Conference on Artificial Intelligence, Epia 2009, Aveiro, Portugal, October 12-15, 2009, Proceedings*. Springer, 2009, p. 423.

[14] Y. Yang, X. Wu, and X. Zhu, "Combining proactive and reactive predictions for data streams," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, R. Grossman, R. Bayardo, and K. P. Bennett, Eds. New York, NY, USA: ACM, 2005, pp. 710–715.

[15] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Tracking recurring contexts using ensemble classifiers: an application to email filtering," *Knowl. Inf. Syst.*, vol. 22, no. 3, pp. 371–391, Mar. 2010.

[16] S. Ramamurthy and R. Bhatnagar, "Tracking recurrent concept drift in streaming data using ensemble classifiers," in *Proceedings of the Sixth International Conference on Machine Learning and Applications*. IEEE Computer Society, 2007, pp. 404–409.

[17] J. B. Gomes, E. Menasalvas, and P. A. C. Sousa, "Learning recurring concepts from data streams with a context-aware ensemble," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, ser. SAC '11. New York, NY, USA: ACM, 2011, pp. 994–999.

[18] P. Turney, "Exploiting context when learning to classify," in *Proceedings of the European Conference on Machine Learning (ECML-93)*, 1993, pp. 402–407.

[19] J. B. Gomes, E. Menasalvas, and P. A. Sousa, "Calds: context-aware learning from data streams," in *Proceedings of the First International Workshop on Novel Data Stream Pattern Mining Techniques*. ACM, 2010, pp. 16–24.

[20] J. B. Gomes, M. M. Gaber, P. A. Sousa, and E. Menasalvas, "Mining recurring concepts in a dynamic feature space," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, 2013.

[21] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the

multiple instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, no. 12, pp. 31–71, 1997.

[22] G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive Online Analysis, 2007 - http://sourceforge.net/projects/moa-datastream/," 2007.

[23] Q. Zhang and S. A. Goldman, "EM-DD: An improved multiple-instance learning technique," in *In Advances in Neural Information Processing Systems*. MIT Press, 2001, pp. 1073–1080.

[24] D. Brzezinski and J. Stefanowski, "Reacting to different types of concept drift: The accuracy updated ensemble algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 81–94, 2013.

[25] J. B. Gomes, M. M. Gaber, P. Sousa, and E. Menasalvas, "Collaborative data stream mining in ubiquitous environments using dynamic classifier selection," *International Journal of Information Technology & Decision Making*, vol. 12, no. 06, 2013.

[26] J. B. Gomes, M. M. Gaber, P. A. Sousa, and E. Menasalvas, "Context-aware collaborative data stream mining in ubiquitous devices," in *Advances in Intelligent Data Analysis X*. Springer Berlin Heidelberg, 2011, pp. 22–33.