

Dealing with Concept Drift and Class Imbalance in Multi-label Stream Classification

Eleftherios Spyromitros-Xioufis¹, Myra Spiliopoulou², Grigorios Tsoumakas¹ and Ioannis Vlahavas¹



¹Department of Informatics, Aristotle University of Thessaloniki, Greece

²Faculty of Computer Science, OVG University of Magdeburg, Germany



TWENTY-SECOND INTERNATIONAL
JOINT CONFERENCE
ON ARTIFICIAL INTELLIGENCE

JULY 16-22, 2011
CCIB
BARCELONA, CATALONIA (SPAIN)
WWW.IJCAI-11.ORG

Multi-label Classification

- Classification of data which can be associated with multiple labels
- Why **more than one** or labels?
 - **Orthogonal labels**
 - Thematic and confidentiality labels in the categorization of enterprise documents
 - **Overlapping labels** typical in news
 - An article about Fukushima could be annotated with {"nuclear crisis", "Asia-Pacific news", "energy", environment}
- Where can multi-label classification be useful?
 - Automated annotation of large object collections for information retrieval, tag suggestion, query categorization,...

Stream Classification

- Classification of instances with the properties of a data stream:
 - Time ordered
 - Arriving continuously and at a high speed
 - Concept drift: gradual or abrupt changes in the target variable over time
- Data stream examples:
 - Sensor data, ATM transactions, e-mails
- Desired properties of stream classification algorithms:
 - Handling infinite data with finite resources
 - Adaption to concept drift
 - Real-time prediction

Multi-label Stream Classification (MLSC)

- The classification of streaming multi-label data
 - Multi-label streams are very common (rss feeds, incoming mail)
- Batch multi-label methods
 - Do not have the desired characteristics of stream algorithms
- Stream classification methods
 - Are designed for single-label data
- Only a few recent methods for MLSC
- Special MLSC challenges (explained next):
 - **Multiple concept drift**
 - **Class imbalance**

Concept Drift

- Types of **concept drift**:
 - **Change in the definition of the target variable**, what is spam today may not be spam tomorrow
 - **Virtual concept drift**, change in prior probability distribution
 - In both cases -> the model needs to be revised
- In multi-label streams
 - Multiple concepts (multiple concept drift)
 - Cannot assume that all concepts drift at the same rate
- A mainstream drift adaptation strategy in single-label streams:
 - Moving window: a window that keeps only the most recently read examples

Class Imbalance in Multi-label Data

- Multi-label data exhibit class imbalance:
 - **Inter-class** imbalance
 - Some labels are much more frequent
 - **Inner-class** imbalance
 - Strong imbalance between the numbers of positive and negative examples
- Imbalance can be exacerbated by virtual concept drift:
 - A label may become extremely infrequent for some time
- Consequences:
 - Very few positive training examples for some labels
 - Decision boundaries are pushed away from the positive class

Single Moving Window (SW) in MLSC

5 instance window

Labels	x_{n-10}	x_{n-9}	x_{n-8}	x_{n-7}	x_{n-6}	x_{n-4}	x_{n-3}	x_{n-2}	x_{n-1}	x_n
λ_1				+			+	+		
λ_2			+							+
λ_3	+	+			+	+				+
λ_4		+	+	+						
λ_5				+	+			+	+	

Most recent instance

Old concept

New /current concept

- Implication of having a common window:
 - Some labels may have only a few or even no positive examples inside the window (λ_2, λ_4) – imbalanced learning situation
- If we increase the window size:
 - Enough positive examples for all labels but risk of including old examples
 - Not necessary for all labels. $\lambda_1, \lambda_3, \lambda_5$ already have enough positive examples

Multiple Windows (MW) Approach for MLSC

- Motivation:
 - More positive examples for training infrequent labels
- We associate each label with **two instance-windows**:
 - One with positive and one with negative examples
- The size of the positive window is fixed to a number n_p which should be:
 - Large enough to allow learning an accurate model
 - Small enough to decrease the probability of drift inside the window
- The size of the negative window n_n is determined using the formula $n_n = n_p/r$ where r has the role of balancing the distribution of positive and negative examples

Multiple Windows (MW) Approach for MLSC

Stream	n	p	n	n	p	p	n	n	n	n	n	n	n	p	n	n	p	n	n	n
Single window										*	*	*	*	*	*	*	*	*	*	*
Multiple Window					*	*						*	*	*	*	*	*	*	*	*

SW : window size = 10, $r = 2/8$ **MW**: $n_p = 4$, $n_n = 6$, $r = 2/3$

- Compared to an equally-sized single window we:
 - Over-sample the positive examples by adding the most recent ones
 - Under-sample the negative examples by retaining only the most recent ones
- The high variance caused by insufficient positive examples in the SW approach is reduced
- There is a possible increase in bias due to the introduction of old positive examples
- Usually small because the negative examples will always be current

Essentially Binary Relevance

- Our method follows the binary relevance (BR) paradigm
 - Transforms the multi-label classification problem into multiple binary classification problems
- Disadvantage
 - Potential label correlations are ignored
- Advantages
 - The independent modeling of BR allows handling the expected differences in frequencies and drift rates of the labels
 - Can be coupled with any binary classifier
 - It can be parallelized to achieve constant time complexity with respect to the number of labels

Incremental Thresholding

- BR can typically output numerical scores for each label
 - Confidence scores are usually transformed into hard 0/1 classifications via an implicit 0.5 threshold
- We use an incremental version of the **PCut** (proportional cut) thresholding method:
- Every n instances (n is a parameter):
 - We calculate for each label a threshold that would most accurately approximate the observed frequency of that label in the last n instances
- The calculated thresholds are used on the next batch of n instances

Datasets

- 3 large textual multi-label datasets (tmc2007,imdb,rcv1v2)
- Bag-of-words representation
 - Boolean vectors for imdb and tmc2007
 - Tf-idf vectors for rcv1v2
- Top features were selected according to $\chi^2_{\max}$ criterion [Lewis et al., 2004]
- rcv1v2 is time ordered - tmc2007 and imdb are static but were treated as streams and were processed in their default order

name	#instances	#features	#labels	#cardinality
tcm2007	28596	500b	22	2.219
imdb	120919	1001b	28	1.999
rcv1v2	804414	500n	103	3.240

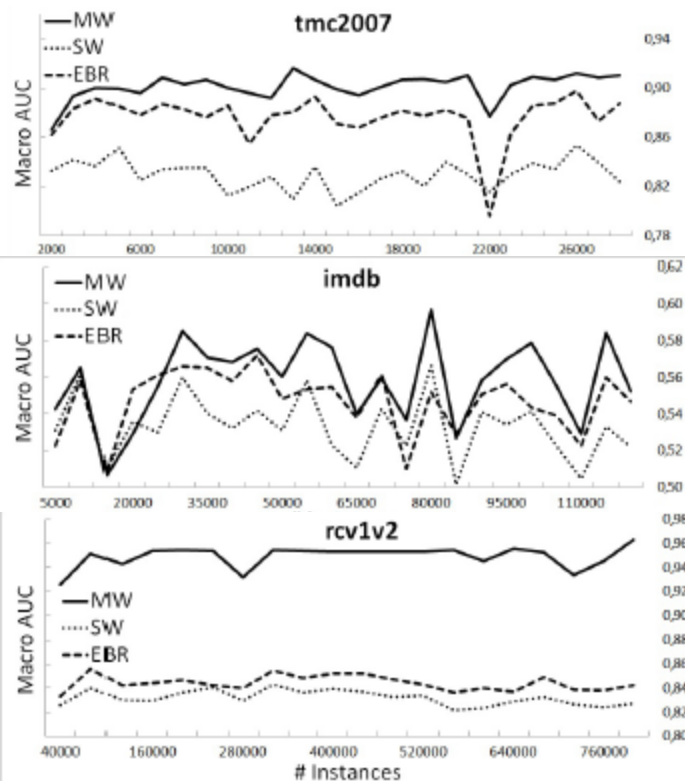
Baselines & Parameter Settings

- MW was compared with:
 - SW: BR operating on a single moving window of N instances
 - EBR: ensemble of k BR classifiers built on k successive chunks
[\[Qu et al., 2009\]](#)
- k-Nearest Neighbors (kNN) was chosen as base classifier for all methods
 - Incremental: incorporates new examples and forgets old examples without need to be rebuilt
 - $k=11$, Jaccard Coefficient as distance function
- All methods were given the same number of training examples for each label

Evaluation Methodology & Measures

- Train-then-test or prequential [[Gama et al., 2009](#)]:
 - An instance is first classified by the current model and then receives its true label and becomes a training example
- Measures:
 - F_1 Measure : The harmonic mean of recall and precision
 - Area Under ROC curve: Calculated directly on the confidence scores, appropriate for threshold independent evaluation
 - Both measures were macro-averaged across all labels

Empirical Results



- Threshold-independent evaluation using macro AUC
- MW is consistently better
- The gains in rcv1v2 are greater (a real data stream – many labels with skewed distributions)

Empirical Results

Dataset	MW		SW		EBR	
	no	th	no	th	no	th
tmc2007	0.415	0.516*	0.313	0.460	0.177	0.461
imdb	0.115	0.131*	0.040	0.116	0.004	0.103
rcv1v2	0.228	0.401*	0.299	0.323	0.071	0.268

- Threshold-dependent evaluation macro F_1
- Substantial gains for all methods when thresholding is applied
- MW is better both with and without thresholding with the exception of rcv1v2 where SW is better when thresholding is not applied
- MW works well with bipartitions when coupled with an appropriate thresholding strategy

Conclusions & Future Work

- Remarks
 - A general framework for dealing with multi-label data streams, independent of the base classifier
 - Space and time efficient implementations were discussed
 - An incremental thresholding method
- Future Work
 - Modeling label correlations (methods like ECC [\[Read et al., 2009\]](#))
 - Employ drift detectors to dynamically adjust the positive window size
 - Experiments with synthetically generated data streams

THANK YOU! QUESTIONS?

See you in the poster session on Friday!

References

- [Gama et al., 2009] J. Gama, R. Sebastiao, and P.P. Rodrigues. Issues in evaluation of stream learning algorithms. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 329–338, 2009.
- [Read et al., 2009] 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.
- [Read et al., 2010] J. Read, A. Bifet, G. Holmes, and B. Pfahringer. Efficient multi-label classification for evolving data streams. Technical Report, April 2010.
- [Qu et al., 2009] W. Qu, Y. Zhang, J. Zhu, and Q. Qiu. Mining Multi-label Concept-Drifting Data Streams Using Dynamic Classifier Ensemble. In Proceedings of the 1st Asian Conference on Machine Learning, pages 308–321, 2009.
- [Lewis et al., 2004] D.D. Lewis, Y. Yang, T.G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. Journal of Machine Learning Research, 5:361–397, December 2004.

Time complexity

- Prediction phase: when used in combination with the kNN algorithm is $O(|B| * |X|)$ where B is the size of the shared buffer and $|X|$ is the number of feature attributes representing each instance.
- Update phase: the complexity is $O(1)$ since kNN requires no training and we just need to update the individual label buffers.

Dataset	Total Update Time (s)	Total Prediction Time (s)	Avg. Prediction Time Per Instance (ms)
tmc2007	1.79	359.28	12
imdb	9.68	2115.83	17
rcv1v2	130.24	31376.08	39

Space complexity

- The space complexity depends on the size of the shared buffer $|B|$
- $|B|$ depends on the following factors:
 1. the size of the positive windows,
 2. the number of observed labels and
 3. the overlap between the labels
- Resource demand for the experiments we reported:

Dataset	Shared buffer size	Memory (mb)
tmc2007	12524	45
imdb	17600	52
rcv1v2	64000	135