Challenges in Learning from Streaming Data
Extended Abstract

João Gama

[1] LIAAD-INESC TEC, University of Porto
[2] Faculty of Economics, University Porto
jgama@fep.up.pt

1 Introduction

Machine learning studies automatic methods for acquisition of domain knowledge with the goal of improving systems performance as the result of experience. In the past two decades, machine learning research and practice has focused on batch learning usually with small data sets. The rationale behind this practice is that examples are generated at random accordingly to some stationary probability distribution. Most learners use a greedy, hill-climbing search in the space of models. They are prone to overfitting, local maximas, etc. Data are scarce and statistic estimates have high variance. A paradigmatic example is the TDIT algorithm to learn decision trees [14]. As the tree grows, less and fewer examples are available to compute the sufficient statistics, variance increase leading to model instability Moreover, the growing process re-uses the same data, exacerbating the overfitting problem. Regularization and pruning mechanisms are mandatory.

The developments of information and communication technologies dramatically change the data collection and processing methods. What distinguish current data sets from earlier ones are automatic data feeds. We do not just have people entering information into a computer. We have computers entering data into each other [7]. Moreover, advances in miniaturization and sensor technology lead to sensor networks, collecting high-detailed spatio-temporal data about the environment.

These technical developments pose new challenges and research oportunities to the data mining community:

- Find the decision structure in the current window;
- What changed in the decision structure last week?
- Which patterns disappeared/appeared last week?
- Which patterns are growing/shrinking this month?
- Mine the evolution of decision structures.

In this paper we review some of the challenges in learning from continuous flow of data.

2 Algorithm Issues in Learning from Data Streams

The challenge problem for data mining is the ability to permanently maintain an accurate decision model. This issue requires learning algorithms that can modify the current

model whenever new data is available at the rate of data arrival. Moreover, they should forget older information when data is out-dated. In this context, the assumption that examples are generated at random according to a stationary probability distribution does not hold, at least in complex systems and for large periods of time. In the presence of a non-stationary distribution, the learning system must incorporate some form of forgetting past and outdated information. Learning from data streams require incremental learning algorithms that take into account concept drift. Solutions to these problems require new sampling and randomization techniques, and new approximate, incremental and decremental algorithms. [9] identify desirable properties of learning systems that are able to mine continuous, high-volume, open-ended data streams as they arrive. Learning systems should be able to process examples and answering queries at the rate they arrive. Some desirable properties for learning in data streams include: incrementality, online learning, constant time to process each example, single scan over the training set, and taking drift into account.

Incremental learning is one fundamental aspect for the process of continuously adaptation of the decision model. The ability to update the decision model whenever new information is available is an important property, but it is not enough, it also require operators with the ability to *forget* past information [13]. Some data stream models allow delete and update operators. Sliding windows models require forgetting old information. In all these situations the incremental property is not enough. Learning algorithms need forgetting operators that reverse learning: decremental unlearning [3].

The incremental and decremental issues requires a permanent maintenance and updating of the decision model as new data is available. Of course, there is a trade-off between the cost of update and the gain in performance we may obtain. Learning algorithms exhibit different profiles. Algorithms with strong variance management are quite efficient for small training sets. Very simple models, using few free-parameters, can be quite efficient in variance management, and effective in incremental and decremental operations being a natural choice in the sliding windows framework. The main problem with simple representation languages is the boundary in generalization performance they can achieve, since they are limited by high bias while large volumes of data require efficient bias management. Complex tasks requiring more complex models increase the search space and the cost for structural updating. These models, require efficient control strategies for the trade-off between the gain in performance and the cost of updating. A step in this direction is the so called *algorithm output granularity* presented by [5]. Algorithm output granularity monitors the amount of mining results that fits in main memory before any incremental integration. [6] illustrate the application of the *algorithm output granularity* strategy to build efficient clustering, frequent items and classification techniques.

In most applications, we are interested in maintaining a decision model consistent with the current status of the nature. This lead us to the sliding window models where data is continuously inserted and deleted from a window. Learning algorithms must have operators for incremental learning and forgetting. Incremental learning and forgetting are well defined in the context of predictive learning. The meaning or the semantics in other learning paradigms (like clustering) are not so well understood, very few works address this issue.

When data flows over time, and at least for large periods of time, it is highly unprovable the assumption that the examples are generated at random according to a stationary probability distribution. At least in complex systems and for large time periods, we should expect changes in the distribution of the examples. A natural approach for these *incremental tasks* are *adaptive learning algorithms*, incremental learning algorithms that take into account concept drift. Concept drift means that the concept related to the data being collected may shift from time to time, each time after some minimum permanence. Changes occur over time. The evidence for changes in a concept are reflected in some way in the training examples. Old observations, that reflect the past behavior of the nature, become irrelevant to the current state of the phenomena under observation and the learning agent must forget that information. The nature of change is diverse. It might occur, in the context of learning, due to changes in hidden variables, or changes in the characteristic properties of the observed variables. Most learning algorithms use blind methods that adapt the decision model at regular intervals without considering whether changes have really occurred. Much more interesting is explicit change detection mechanisms. The advantage is that they can provide meaningful description (indicating change-points or small time-windows where the change occurs) and quantification of the changes. The main research issue is how to incorporate change detection mechanisms in the learning algorithm, embedding change detection methods in the learning algorithm is a requirement in the context of continuous flow of data. The level of *granularity* of decision models is a relevant property, because if can allow partial, fast and efficient updates in the decision model instead of rebuilding a complete new model whenever a change is detected. The ability to recognize seasonal and re-occurring patterns is an open issue.

Novelty detection refers to learning algorithms being able to identify and learn new concepts. Intelligent agents that act in dynamic environments must be able to learn conceptual representations of such environments. Those conceptual descriptions of the world are always incomplete, they correspond to what it is *known* about the world. This is the *open* world assumption as opposed to the traditional *closed* world assumption, where what is to be learnt is defined in advance. In open worlds, learning systems should be able to extend their representation by learning new concepts from the observations that do not match the current representation of the world. This is a difficult task. It requires to identify the *unknown*, that is, the limits of the current model. In that sense, the *unknown* corresponds to an *emerging pattern* that is different from *noise*, or *drift* in previously known concepts.

Data streams are distributed in nature. Learning from distributed data, we need efficient methods in minimizing the communication overheads between nodes [15]. The strong limitations of centralized solutions is discussed in depth in [10, 11]. The authors point out *a mismatch between the architecture of most off-the-shelf data mining algorithms and the needs of mining systems for distributed applications*. Such mismatch may cause a bottleneck in many emerging applications, namely hardware limitations related to the limited bandwidth channels. Most important, in applications like monitoring, centralized solutions introduce delays in event detection and reaction, that can make mining systems useless. Another direction, for distributed processing, explore multiple models [4, 12]. [12] propose a method that offer an effective way to construct a

redundancy-free, accurate, and meaningful representation of large decision-tree ensembles often created by popular techniques such as Bagging, Boosting, Random Forests and many distributed and data stream mining algorithms.

In some challenging applications of Data Mining, data are better described by sequences (for example DNA data), trees (XML documents), and graphs (chemical components). Tree mining in particular is an important field of research [1, 2]. XML patterns are tree patterns, and XML is becoming a standard for information representation and exchange over the Internet; the amount of XML data is growing, and it will soon constitute one of the largest collections of human knowledge.

In the static case, similar data can be described with different schemata. In the case of dynamic streams, the schema of the stream can also change. For example, in monitoring sensor networks, and social network analysis, new nodes might appear and others might diappear. We need algorithms that can deal with evolving feature spaces over streams. There is very little work in this area, mainly pertaining to document streams. For example, in sensor networks, the number of sensors is variable (usually increasing) over time.

An important aspect of any learning algorithm is the hypothesis evaluation criteria. Most of evaluation methods and metrics were designed for the static case and provide a single measurement about the quality of the hypothesis. In the streaming context, we are much more interested in how the evaluation metric evolves over time. Results from the *sequential statistics* [16] may be much more appropriate. [8] propose a general framework for assessing predictive stream learning algorithms using sequential statistics. They show that the prequential error converges to an holdout estimator when computed over sliding windows or using fading factors.

3 Conclusions

The ultimate goal of Data Mining is to develop systems and algorithms with high level of autonomy. For such, Data Mining studies the automated acquisition of domain knowledge looking for the improvement of systems performance as result of experience. These systems address the problems of data processing, modeling, prediction, clustering, and control in changing and evolving environments. They self-evolve their structure and knowledge on the environment.

The challenges and research opportunities of data streaming mining are abundant. It is one of most pleasent research areas nowadays.

Acknowledgments

This work was supported by Sibila research project (NORTE-07-0124-FEDER-000059), financed by North Portugal Regional Operational Programme (ON.2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT), and by European Commission through the project MAESTRA (Grant number ICT-2013-612944).

References

1. Albert Bifet and Ricard Gavaldà. Mining adaptively frequent closed unlabeled rooted trees in data streams. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, pages 34–42, Las Vegas, USA, 2008.
2. Albert Bifet and Ricard Gavaldà. Adaptive XML tree classification on evolving data streams. In *Machine Learning and Knowledge Discovery in Databases, European Conference*, volume 5781 of *Lecture Notes in Computer Science*, pages 147–162, Bled, Slovenia, 2009. Springer.
3. Gert Cauwenberghs and Tomaso Poggio. Incremental and decremental support vector machine learning. In *Proceedings of the Neural Information Processing Systems*, 2000.
4. R. Chen, K. Sivakumar, and H. Kargupta. Collective mining of Bayesian networks from heterogeneous data. *Knowledge and Information Systems Journal*, 6(2):164–187, 2004.
5. M. Gaber, M and Yu P. S. A framework for resource-aware knowledge discovery in data streams: a holistic approach with its application to clustering. In *ACM Symposium Applied Computing*, pages 649–656. ACM Press, 2006.
6. Mohamed Medhat Gaber, Shonali Krishnaswamy, and Arkady Zaslavsky. Cost-efficient mining techniques for data streams. In *Proceedings of the second workshop on Australasian information security*, pages 109 – 114. Australian Computer Society, Inc., 2004.
7. Joao Gama. *KnowledgeDiscovery from Data Streams*. Data Mining and Knowledge Discovery. Chapman & Hall CRC Press, Atlanta, US, 2010.
8. João Gama, Raquel Sebastião, and Pedro Pereira Rodrigues. Issues in evaluation of stream learning algorithms. In *KDD*, pages 329–338, 2009.
9. Geoff Hulten and Pedro Domingos. Catching up with the data: research issues in mining data streams. In *Proc. of Workshop on Research Issues in Data Mining and Knowledge Discovery*, Santa Barbara, USA, 2001.
10. H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha. *Data Mining: Next Generation Challenges and Future Directions*. AAAI Press and MIT Press, 2004.
11. Hillol Kargupta and Byung-Hoon Park. Mining decision trees from data streams in a mobile environment. In *IEEE International Conference on Data Mining*, pages 281–288, San Jose, USA, 2001. IEEE Computer Society.
12. Hillol Kargupta, Byung-Hoon Park, and Haimonti Dutta. Orthogonal decision trees. *IEEE Transactions on Knowledge and Data Engineering*, 18:1028–1042, 2006.
13. Daniel Kifer, Shai Ben-David, and Johanes Gehrke. Detecting change in data streams. In *Proceedings of the International Conference on Very Large Data Bases*, pages 180–191, Toronto, Canada, 2004. Morgan Kaufmann.
14. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, Inc. San Mateo, CA, 1993.
15. Izchak Sharfman, Assaf Schuster, and Daniel Keren. A geometric approach to monitoring threshold functions over distributed data streams. *ACM Transactions Database Systems*, 32(4):301–312, 2007.
16. A. Wald. *Sequential Analysis*. John Wiley and Sons, Inc, 1947.