Issues and Challenges in Learning from Data Streams
Extended Abstract

João Gama
LIAAD-INESC-Porto LA, and FEP-University of Porto
R. de Ceuta 118-6; 4050 Porto-Portugal
jgama@fep.up.pt

1 Introduction

In the last two decades, machine learning research and practice has focused on batch learning usually with small datasets. In batch learning, the whole training data is available to the algorithm, that outputs a decision model after processing the data eventually (or most of the times) multiple times. 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.

The development of information and communication technologies dramatically change the data collection and processing methods. Advances in miniaturization and sensor technology lead to sensor networks, collecting detailed spatio-temporal data about the environment.

An illustrative application is the problem of mining data produced by sensors distributed all around electrical-power distribution networks. These sensors produce streams of data at high-speed. From a data mining perspective, this problem is characterized by a large number of variables (sensors), producing a continuous flow of data, in a dynamic non-stationary environment. Companies analyze these data streams and make decisions for several problems. Companies are interested in identify critical points in load evolution, e.g. picks on the demand. These aspects are related to anomaly detection, extreme values, failures, outliers, and abnormal activities detection. Other problem is related to change detection in the behavior (correlation) of sensors. Cluster analysis can be used for the identification of groups of high-correlated sensors, corresponding to common behaviors or profiles (e.g. Urban, Rural, Industrial, etc.). Decisions to buy or sell energy are based on the predictions on the value measured by each sensor for different time horizons. All these problems illustrates some of the requirements and objectives usually inherent to ubiquitous computing. Sensors produce a continuous flow of data, are limited in resources such as memory and computational power, and communication between them is easily narrowed due to distance and hardware limitations. Moreover, given the limited resources and fast production of data, information must be processed in real-time, creating a scenario of multi-dimensional streaming analysis.

In this article we discuss the issues and challenges on learning from data streams. We discuss limitations of current learning systems and point out possible research lines for next generation data mining systems. How to learn from these distributed continuous streaming data? Which are the main characteristics of a learning algorithm acting in sensor networks? What are the relevant issues, challenges, and research opportunities?

2 Machine Learning and Data Streams

What distinguishes current data sets from earlier ones are the continuous flow of data and the automatic data feeds. We do not just have people who are entering information into a computer. Instead, we have computers entering data into each other (Muthukrishnan, 2005). Nowadays there are applications in which the data is modeled best not as persistent tables but rather as transient data streams. In some applications it is not feasible to load the arriving data into a traditional DataBase Management Systems (DBMS), and traditional DBMS are not designed to directly support the continuous queries required in these applications (Babcock et al., 2002).

2.1 Streaming Algorithms

In the streaming model (Muthukrishnan, 2005) the input elements $a_1, a_2, \ldots, a_j, \ldots$ arrive sequentially, item by item and describe an underlying function A. From the view point of a data streams management system, several research issues emerge. One relevant issue is approximate query processing techniques to evaluate continuous queries that require unbounded amount of memory. Sampling has been used to handle situations where the flow rate of the input stream is faster than the query processor. Another relevant issue is the definition of the semantics (and implementation) of blocking operators (operators that only return an output tuple after processing all input tuples, like aggregation and sorting) in the presence of unending streams.

Algorithms that process data streams deliver approximate solutions, providing a fast answer using few memory resources. They relax the requirement of an exact answer to an approximate answer within a small error range with high probability. In general, as the range of the error decreases the space of computational resources goes up. In some applications, mostly database oriented, an approximate answer should be within an admissible error margin. Data Streams Management Systems developed a set of techniques that store compact stream summaries enough to approximately solve queries. All these approaches require a trade-off between accuracy and the amount of memory used to store the summaries, with an additional constrain of small time to process data items (Muthukrishnan, 2005). The most common problems end up to compute quantiles, frequent item sets, and to store frequent counts along with error bounds on their true frequency. The techniques developed in data streams management systems can provide tools for designing machine learning algorithms in very high dimensions both in the number of examples and in the cardinality of the variables. On the other hand, machine learning provides compact descriptions of the data than can be useful for answering queries in DSMS.

2.2 Algorithm Issues: Online, Anytime and Real-time Learning

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. Hulten and Domingos (2001) 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.

2.2.1 Incremental and Decremental issues

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. Another required operator is the ability to *forget* past information (Kifer et al., 2004). 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 (Cauwenberghs and Poggio, 2000).

2.2.2 Cost-Performance Management

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 (for example naive Bayes) 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. 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.

2.3 Monitoring Learning

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.

2.3.1 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. Changes may occur in the context of learning, due to changes in hidden variables, or in the characteristic properties of the observed variables.

2.3.2 Methods and Algorithms for Change Detection

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. They may follow two different approaches:

1. Monitoring the evolution of performance indicators adapting techniques used in Statistical Process Control.

2. Monitoring distributions on two different time windows.

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.

2.4 Novelty Detection

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.

2.5 Algorithm Issues in Learning from Data Streams

Streaming data and domains offer a nice opportunity for a symbiosis between Streaming Data Management Systems and Machine Learning. The techniques developed to estimate synopsis and sketches require counts over very high dimensions both in the number of examples and in the domain of the variables. The techniques developed in data streams management systems can provide tools for designing Machine Learning algorithms in these domains. On the other hand, Machine Learning provides compact descriptions of the data than can be useful for answering queries in DSMS.

Incremental Learning and Forgetting. 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.

Change Detection. *Concept drift* in the predictive classification setting is a well studied topic. In other learning scenarios, like clustering, very few works address the problem. The main research issue is how to incorporate change detection mechanisms in the learning algorithm for different paradigms.

Feature Selection and Pre-processing. Selection of relevant and informative features, discretization, noise and rare events detection are common tasks in Machine Learning and Data Mining. They are used in a one-shot process. In the streaming context the semantics of these tasks changes drastically. Consider the feature selection problem. In streaming data the concept of *irrelevant* or *redundant* features are now restricted to a certain period of time. Features previously considered *irrelevant* may become *relevant*, and vice-versa to reflect the dynamics of the process generating data. While in standard data mining, an irrelevant feature could be ignored forever, in the streaming setting we need still monitor the evolution of those features. Recent work based on the *fractal dimension* (Barbara and Chen, 2000) could point interesting directions for research.

Evolving Feature Spaces. 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. 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.

Evaluation Methods and Metrics. 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* (Wald, 1947) may be much more appropriate.

There is a fundamental difference between learning from small datasets and large datasets. As pointed-out by some researchers (Brain and Webb, 2002), current learning algorithms emphasize variance reduction. However, learning from large datasets may be more effective when using algorithms that place greater emphasis on bias management.

3 Emerging Challenges and Future Issues

Simple objects that surround us are changing from static, inanimate objects into adaptive, reactive systems with the potential to become more and more useful and efficient. Smart things associated with all sort of networks offers new unknown possibilities for the development and self-organization of communities of intelligent communicating appliances. The dynamic characteristics of data flowing over time requires adaptive algorithms. While the languages used to represent generalizations from examples are well understood, next generation data mining algorithms should care, at least, about the cost-performance management; the limitations in working in an open-world that implies limitations in the knowledge about the learning goals and the limitations in all aspects of computational resources. All these aspects requires monitoring the evolution of learning process itself, and the ability of reasoning and learning about it.

References

Babcock, B., Babu, S., Datar, M., Motwani, R., and Widom, J. (2002). Models and issues in data stream systems. In Kolaitis, P. G., editor, *Proceedings of the 21nd Symposium on Principles of Database Systems*, pages 1–16. ACM Press.

Barbara, D. and Chen, P. (2000). Using the fractal dimension to cluster datasets. In *Proc. of the 6th International Conference on Knowledge Discovery and Data Mining*, pages 260–264. ACM Press.

Brain, D. and Webb, G. (2002). The need for low bias algorithms in classification learning from large data sets. In T.Elomaa, H.Mannila, and H.Toivonen, editors, *Principles of Data Mining and Knowledge Discovery PKDD-02*, pages 62–73. LNAI 2431, Springer Verlag.

Cauwenberghs, G. and Poggio, T. (2000). Incremental and decremental support vector machine learning. In *Proceedings of the 13th Neural Information Processing Systems*.

Hulten, G. and Domingos, P. (2001). Catching up with the data: research issues in mining data streams. In *Proc. of Workshop on Research issues in Data Mining and Knowledge Discovery*.

Kargupta, H., Joshi, A., Sivakumar, K., and Yesha, Y. (2004). *Data Mining: Next Generation Challenges and Future Directions*. AAAI Press and MIT Press.

Kifer, D., Ben-David, S., and Gehrke, J. (2004). Detecting change in data streams. In *VLDB 04: Proceedings of the 30th International Conference on Very Large Data Bases*, pages 180–191. Morgan Kaufmann Publishers Inc.

Motwani, R. and Raghavan, P. (1997). *Randomized Algorithms*. Cambridge University Press.

Muthukrishnan, S. (2005). *Data streams: algorithms and applications*. Now Publishers.

Oza, N. (2001). *Online Ensemble Learning*. PhD thesis, University of California, Berkeley.

Wald, A. (1947). *Sequential Analysis*. John Wiley and Sons, Inc.