Information Sciences 379 (2017) 313–337

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Ranked batch-mode active learning

Thiago N.C. Cardoso, Rodrigo M. Silva, Sérgio Canuto, Mirella M. Moro, Marcos A. Gonçalves*

Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Brazil

ARTICLE INFO

Article history:
Received 3 February 2016
Revised 6 October 2016
Accepted 22 October 2016
Available online 28 October 2016

Keywords:
Active learning
Batch-mode
New ranking function
Paradigm change

ABSTRACT

We introduce a new paradigm for *Ranked Batch-Mode Active Learning*. It relaxes traditional Batch-Mode Active Learning (BMAL) methods by generating a query whose answer is an **optimized ranked list** of instances to be labeled, according to some quality criteria, allowing batches to be of arbitrarily large sizes. This new paradigm avoids the main problem of traditional BMAL, namely the frequent stops for manual labeling, reconciliation and model reconstruction. In this article, we formally define this problem and introduce a framework that iteratively and effectively builds the ranked list. Our experimental evaluation shows our proposed Ranked Batch approach significantly reduces the number of algorithm executions (and, consequently, the manual labeling delays) while maintaining or even improving the quality of the selected instances. In fact, when using only unlabeled data, our results are much better than those produced by pool-based batch-mode active learning methods that rely on already labeled seeds or update their models with labeled instances, with gains of up to 25% in MacroF1. Finally, our solutions are also more effective than density-sensitive active learning methods in most of the envisioned scenarios, as demonstrated by our experiments.

1. Introduction

For many applications, the volume of data generated every day is huge and impracticable to be manually read and processed by human analysts. Analyzing such huge volume of data is also impractical without automated systems. In this context, machine learning (ML) algorithms have become increasingly popular for automatically treating large volumes of data.

Supervised ML methods, which historically have produced the best results in the literature, extract important patterns derived from a dataset labeled by human experts, and apply those patterns to unseen data to perform the desired task. However, labeling the training examples is usually expensive because of the necessary expert knowledge and the time-consuming nature of the process. Moreover, a large number of labeled instances is often required in order to achieve an acceptable error rate. Finally, constant training data is also necessary to cope with changes on patterns of use.

A reduction in the effort of creating such training sets has motivated the introduction of *Active Learning* (AL) [1]. This type of technique selects and presents to the analyst/expert (also called *oracle*) instances that should be labeled first based on an estimate of the gain of information they can bring to the overall learning process. This group of instances is called

* Corresponding author.
E-mail addresses: thiagon@dcc.ufmg.br (T.N.C. Cardoso), rmsilva@dcc.ufmg.br (R.M. Silva), sergiodaniel@dcc.ufmg.br (S. Canuto), mirella@dcc.ufmg.br (M.M. Moro), mgoncalv@dcc.ufmg.br (M.A. Gonçalves).

http://dx.doi.org/10.1016/j.ins.2016.10.037
0020-0255/

a *query* since it requires answers (i.e., labels) from the oracle[1]. After labeling, such instances are incorporated into the training set with an expectation of rapidly increasing the classifier effectiveness. In general, traditional AL algorithms query one instance at a time in order to update the classification model. Such a limitation can make the manual labeling routine very daunting and inadequate in scenarios with multiple experts working simultaneously in the same dataset, or when the query construction requires an expensive process implying in a long waiting time. For instance, creating a labeled set using a crowdsourcing service like Amazon Mechanical Turk[2] is impractical with such limitations. Also, such crowdsourcing services allow users to create labeling tasks that are outsourced to a large group of freelance workers; therefore, querying one instance at a time takes no advantage of such highly parallel environment.

To alleviate the aforementioned problems, a *Batch-Mode* AL (BMAL) algorithm can be used [2–4]. This class of algorithms query multiple instances at once and such instances can be independently labeled in parallel. However, BMAL by itself is still a time consuming task due to the practical problem of organizing and orchestrating the actual real-world labeling process. Specifically, the fundamental issue in using a BMAL solution lies in organizing the labeling effort efficiently. Once a batch of unlabeled samples is produced, its instances are distributed to the human analysts for labeling, which requires waiting for all of them to finish labeling the batch before these instances can be incorporated into the training set. Only then new learning models can be produced[3] and a new batch of unlabeled samples created (this process goes on and on in this interactive manner).

Moreover, the nature of the dataset/labeling task may require the design of a labeling process that provides some protection from labeling noise, such as having a single instance labeled by several analysts and then checking for inconsistencies in the labels produced by different analysts. This process alone can take many hours if not days (each analyst can tackle the task assigned to him/her within a time frame, and the essentially asynchronous nature of this process over the span of several iterations generates the biggest delays). Once *all* responses are in, we must analyze them following the agreed-upon protocol to determine the final labeling of each instance in the batch. Only then the production of another batch may begin, starting this cumbersome process all over again.

Overall, before building a new batch, important latency points do exist: (i) analyst labeling; (ii) labeling agreement; and (iii) model reconstruction (i.e., learning a new model with the newly labeled batch incorporated into training). This process will be repeated dozens or hundreds of times. The essential problem is that every batch generates a series of new *tasks*, including those to be distributed to the analysts, which must be finished (i.e., waited for) before producing new ones.

Simply increasing the batch size of a batch-mode AL method is not a good option either. Batch-mode methods will produce worse results as the batch size increases, simply because that is not how they were designed to operate. These algorithms *heavily rely* on having many iterations with small batches where the learner model is improved and adapted little by little. Hence, they will not perform well when this process is changed to the worst scenario where the batch becomes larger or in the extreme case, comprises the complete unlabeled set.

To overcome these issues, we introduce the *Ranked Batch-Mode Active Learning* (RBMAL) problem. The main idea is to relax some of the aforementioned limitations such that the query is a *ranked* list according to some quality criteria, instead of an unordered set. The problem is well connected to real world scenarios in which either analysts are paid hourly to build a training set for a given problem or the gains must be maximized on a limited budget. Our new approach allows the algorithm to generate an arbitrarily long query, thus making its execution less frequent. For example, a ranked query containing every available instance could be generated outside working hours. This would allow hired analysts to label instances for a full day, in parallel, if desired, without waiting for the learner update and query reconstruction.

Accordingly, in here we propose a method for ranking an arbitrarily large set of unlabeled instances in descending order of informativeness; that is, the order in which they should be labeled by the oracle, based on the current information provided as input and/or produced by the method at a given point of its execution. In other words, after one or more instances are labeled, the acquired knowledge can be incorporated by the method and used to generate a new, more accurate, ranking of the (still) unlabeled instances.

The main contributions of this work can be summarized as follows:

- We advocate a new and different way of thinking about the BMAL problem (what we call ranked batch-mode active learning – RBMAL), by directly optimizing the ranking of instances to be generated for labeling, instead of focusing on *iterative* small batches of instances necessary for the proper working of BMAL methods. Such rereading of the problem solves some of the drawbacks of current methods (Section 3).
- We propose a framework for solving the problem by building a ranked query even when no labeled data is provided (Section 4). This is performed by *locally and dynamically* optimizing the ranking under creation based on the information currently available on it.
- We propose a simple, yet effective solution for tackling the cold start problem (when there is no initial training data) (Section 5).

[1] Note that this notion of query is very different from the one commonly used in Information Retrieval and similar fields.

[2] Amazon Mechanical Turk https://www.mturk.com .

[3] Notice that model reconstruction is a basic premise of BMAL solutions as new batches will be based on how much information it will bring to the updated model. This process itself can incur in considerable latency due to issues such as large datasets, complex numerical optimizations and parameter tuning.