Investigating the Benefits of Exploiting Incremental Learners Under Active Learning Scheme

Stamatis Karlos, Vasileios G. Kanas, Nikos Fazakis, Christos Aridas, Sotiris Kotsiantis

Investigating the benefits of exploiting incremental learners under Active Learning scheme

Stamatis Karlos[1][0000-0002-5307-6186], Vasileios G. Kanas[2], Nikos Fazakis[2], Christos Aridas[1] and Sotiris Kotsiantis[1]

[1] University of Patras, Department of Mathematics, Rio Campus, 26504, Greece
[2] University of Patras, Department of Electrical & Computer Engineering, Rio Campus, 26504, Greece
stkarlos@upatras.gr, vaskanas@upatras.gr, fazakis@ece.upatras.gr, char@upatras.gr, sotos@math.upatras.gr

Abstract. This paper examines the efficacy of incrementally updateable learners under the Active Learning concept, a well-known iterative semi-supervised scheme where the initially collected instances, usually a few, are augmented by the combined actions of both the chosen base learner and the human factor. Instead of exploiting conventional batch-mode learners and refining them at the end of each iteration, we introduce the use of incremental ones, so as to apply favorable query strategies and detect the most informative instances before they are provided to the human factor for annotating them. Our assumption about the benefits of this kind of combination into a suitable framework is verified by the achieved classification accuracy against the baseline strategy of Random Sampling and the corresponding learning behavior of the batch-mode approaches over numerous benchmark datasets, under the pool-based scenario. The measured time reveals also a faster response of the proposed framework, since each constructed classification model into the core of Active Learning concept is built partially, updating the existing information without ignoring the already processed data. Finally, all the conducted comparisons are presented along with the appropriate statistical testing processes, so as to verify our claim.

Keywords: Incremental learners, Active Learning scheme, Stochastic Gradient Descent, Query strategy, unlabeled data.

1 Introduction

Today, more and more applications from various scientific domains produce large volumes of data, changing the needs of current predictive mechanisms that mainly stem from the Machine Learning (ML) field. Since time and memory constitute the two main factors that highly define the performance of intelligent algorithms, especially when they tackle with problems over the era of Big Data, data scientists and ML/data engineers have to prioritize the structure of new predictive tools according to these specifications [1]. Incremental learning is the answer of the ML community to such kind of issues, where the principal idea is to update an existing or a previously built learning

model by exploiting the newly available data, reducing the total time demands while possibly producing less accurate models [2].

Besides the simple approach, according to which vast amounts of labeled data (L) are provided or are reaching into data streams, a more realistic scenario has to cope with the shortage of L, in contrast with high enough volumes of unlabeled data (U). One representative reason why this may happen is the fact that in several real-world applications (e.g. in medicine tasks or in long-term experiments) the final state of the target variable may demand large time periods to be verified or to converge. Another reason is the inherit complexity of the data. In case of text-mining and nature language processing, numerous articles, chapters and posts on social media are freely available through web. However because of the complex structure that may characterize these sources – such as the complicated or unexplored semantic meanings on languages other than English – neither the automated solutions produce always decent learning results nor the choice of manually scanning by human entities could be time efficient [3].

In order to handle this phenomenon, a new kind of algorithms has been raised, often called as Semi-Supervised Learning (SSL) or, even more generic, as Partially-Supervised Learning (PSL), where the former category is contained into the latter [4]. The ambition of PSL algorithms is to exploit the existing labeled instances (l_i) along with the collected unlabeled examples (u_i) and construct a model that maps the unknown instances with the target variable better than the corresponding model, which is based exclusively on the L subset. One main division among PSL algorithms depends on the way that the corresponding u_i are getting labeled before they are merged into the *L* subset, so as to contribute over the increase of the predictive performance of the whole algorithm. While in SSL algorithms this process is automated usually by a base learner, Active Learning (AL) algorithms are differentiated since a human oracle is inserted into the learning process and is responsible for assigning the selected by a criterion u_i with accurate enough decisions [5]. Although in several domains, only human experts could be exploited, it has been studied and generally verified that the decisions of numerous simple-users tend to converge over these produced by the human specialists in domains like sound/music signal categorization [6]. This means that a large aspect of applications could be satisfied through AL approaches without consuming much humans' expert effort, a fact that might convert this kind of solution into a non-affordable one.

Our ambition in this work is to investigate the benefits of exploiting incremental learners (IncL) under a simple AL scheme that follows pool-based scenario. Thus, IncL would be responsible for detecting the most suitable u_i, as well as for exporting the final decisions. The main rivals here are both the AL method whose query strategy coincides with a random selection of u_i and the supervised scenario, where the same base learner has been trained based on the full dataset incrementally, as well as the same approaches trained under batch-mode operation. The amount of the initial *L* plays a crucial role producing per each different value a new variant of each exported algorithm from the proposed framework. More comments are presented in the corresponding paragraph.

To sum up, in Section 2, a number of related works are presented briefly, regarding mainly recent publications over incremental learning task and secondly with AL. Section 3 contains a description of the selected optimizer that injects the desired asset of

incrementally update over underlying linear classifiers along with the necessary information about the proposed AL framework, while Section 4 includes more technical information, as well as information about the examined datasets and the conducted experiments. Finally, this work finishes with the conclusory Section that discusses the posed ambitions and highlights future work.

2 Related work

2.1 Incremental learning

Incremental learning refers to online learning strategies applicable to real-life streaming scenarios [7] with limited memory resources. So far, IncL is widely used ranging from Big Data and Internet of Things technology [7] to outlier detection for surveillance systems. One of the most popular areas in IncL is image/video data processing. Typical case scenarios are object detection [8] and recognition [9], image segmentation [10] and classification [11], surveillance [12], visual tracking [13] and prediction [14].

Moreover, the inherit nature of data in robotics make online learning an appropriate approach for mining the streaming signals [15]. In another study [16] an incremental image semantics learning framework is proposed. The proposed framework aims to learn image semantics from scratch (without a priory knowledge) and enrich the knowledge incrementally with human-robot interactions based on a teaching-and-learning procedure. Khan et al [17] present a mechanism to build a consistent topological map for self-localization robotics applications. The proposed appearance-based loop closure detection mechanism builds a binary vocabulary consisting of visual words in an online, incremental manner by tracking features between consecutive video frames to incorporate pose invariance. In another relevant study [18], authors propose a new method to incrementally learn end-effector and null-space motions via kinesthetic teaching allowing the robot to execute complex tasks and to adapt its behavior to dynamic environments. The authors combine IncL with a customized multi-priority kinematic controller to guarantee a smooth human-robot interaction. Another rapidly emerging domain, which utilizes this concept, is robotic automotive [19].

2.2 Active learning

However, all of the above applications require lots of labelled data and usually data labelling is difficult, time-consuming, and/or expensive to obtain. Active learning systems attempt to overcome the labeling bottleneck by asking queries in the form of u_i to be labeled by an oracle, aiming to significantly reduce the cardinality of L subset that is needed. Active learning is still being heavily researched, under either more theoretical approaches or more experimental ones.

The last years, several attempts have been made to combine AL with Deep Learning (DL) concept, especially targeting specific applications that demand much computational power. Hence, ML researchers have begun searching the benefits of using CNNs and LSTMs and how to improve their efficiency when are applied along with AL frame-

works [20, 21]. There is also research being done on implementing Generative Adversarial Networks (GANs) into the this kind of tasks [22]. With the increasing interest into deep reinforcement learning, researchers are trying to reframe AL as such a problem [23]. Also, there are papers which try to learn AL strategies via a meta-learning setting [24]. This does not mean that products of ML or more simple probabilistic base learners have been ignored by the corresponding community. On the contrary, a recent demonstration examines the chance of achieving fast and non-myopic AL strategy in context of binary classification datasets [25].

3 Proposed Framework

In order to conduct our investigation, a series of properties have to be defined for formulating the corresponding framework, under which our experiments will be executed. To be more specific, the base learner that is used in the core of the proposed framework is based on regularized linear models manipulated by Stochastic Gradient Descent (SGD) learning [26]. A more in-depth analysis follows subsequently, into this Section.

The same learner is used into both the selected Query strategy of AL framework and the stage of building the final classifier, after having augmented the *L* subset during each one of the *k* executed iterations. As it concerns the Query strategy, Uncertainty Sampling (UncS) approach has been selected in the context of this work, favoring the integration of the AL framework with probabilistic classifiers and boosting also the time response of each produced approach [27], [28]. Analog to the metric that is applied into the UncS approach, a number of variants can be produced. Finally, the human factor is replaced by an ideal oracle (H^{oracle}) that exports always the correct decision about the label of each asked instance, playing the role of annotator.

The last generic parameter that has to be set is the Labeled Ratio value – usually depicted as *R* – and measured in percentage values. This factor defines the amount of the initially l_i in comparison with the total amount of both l_i and u_i. Its formula is:

$$R\ (\%) = cardinality(L) / (cardinality(L) + cardinality(U)) \quad (1)$$

It is prominent that by acting under small *R* values, only a small part of the totally available information is provided initially to the AL framework. Hence, the predictions of the base learner are based on poor *L* subsets that may not reveal useful insights of the specific problem that is tackled, harnessing the achievement of accurate classification behaviors. Thus, the quadruple that defines each product of the proposed framework consists of the base learner, the specific metric of UncS, the number of iterations and the Labeled Ratio value. Its notation hereinafter would be (base-cl, $UncS_{metric}$, k, R). The obtained learning behavior of such an algorithm, according to our assumptions, would depict the ability of the selected base learner to operate efficiently under a fast and confident Query strategy to choose among a pool of u_i, over which will be trained incrementally for *k* iterations, before exporting a final classifier, based initially on an amount of labeled instances that is defined by R parameter.

Gradient descent (GD) is by far the most popular optimization strategy, used in ML and DL at the moment. It is an optimization algorithm, based on a convex function, that

tweaks its parameters iteratively to minimize a given cost function to its local minimum. In a simple supervised learning setup, each training example is composed of an arbitrary input *x* and a scalar output *y* in the form (x, y). For our ML model, we choose a family *G* of functions such as $y \cong g_w(x) + b$, with *w* being a weighted vector and *b* an intercept term, which is necessary for obtaining better fit. Consequently, our goal is to minimize a cost function $\Psi(\hat{y}, y) = \Psi(g_w(x), y)$ that measures the cost of predicting $\hat{y}$ given the actual outcome y (or y_{actual}) averaged on the training examples *n*. In other words, we seek to find a solution to the following problem:

$$E_n(f_w, w) = \frac{1}{n}\sum_{j=1}^{n} l\left(f_w(x_j), y_j\right) + \alpha Reg(w) \quad (2)$$

The cost function (E_n) of Eq. 2 depends mainly on loss function (*l*) and the regularization term $Reg(w)$. The multiplicative constant α refers to a non-negative hyperparameter. Following the original GD process, the minimization of Eq. 2 is taking place updating the next two formulas per each iteration t:

$$w_{t+1} = w_t - \eta\left(\frac{1}{n}\sum_{j=1}^{n} \nabla_w l\left(f_w(x_j), y_j\right) + a\nabla_w Reg(w)\right) \quad (3)$$

where the positive scalar η is called the learning rate or step size. In order, for the algorithm, to achieve linear convergence sufficient regularity assumptions should be made, while the initial estimate w_0 should be close enough to the optimum and the gain η sufficiently small. It is important to highlight that the evaluation of n derivatives is required at each step. So, the per-iteration computational cost scales linearly with the training data set size n, making the algorithm inapplicable to huge datasets. Thus, the stochastic (SGD) version of the algorithm is used instead, which offers a lighter-weight solution. More specifically, at each iteration, the SDG randomly picks an example and calculates the gradient for this specific example:

$$w_{t+1} = w_t - \eta_t(\nabla_w l(f_w(x_t), y_t) + a\nabla_w Reg(w)) \quad (5)$$

In other words, SGD approximates the actual gradient using only one data point, saving a lot of time compared to summing over all data. SGD often converges much faster compared to GD but the error function is not as well minimized as in the case of GD. However, in most cases, the close approximation calculated by SGD for the parameter values are enough because they reach the optimal values and keep oscillating there. Another advantage of SGD is its ability to process the incoming data online in a deployed system, since no memory of the previous randomly chosen examples is necessary. In such a situation, the SGD directly optimizes the expected risk, since the examples are randomly drawn from the ground truth distribution [29].

As it concerns $Reg(w)$ term, three different choices are generally used in the literature: l1 norm that favors sparse solutions, l2 norm that is the most usual met and elastic net (elnet), that is formatted by a convex combination of the previous two norms and offers sparsity with better stabilization than simple l1 norm [30]. Before introducing the proposed framework through suitable pseudocode, we have to define the amount of the u_i examples that should be mined per iteration. Although many approaches prefer to mine only one example per time, leading probably to more accurate actively trained

classifier but clearly demanding much more computational resources because of the large amount of iterations that should be executed under a specific budget plan (B), a heuristic method is applied here: the questioned quantity of mined instances per iteration is computed by dividing the initial size of *L* subset with the number of executed iterations *k*. Thus, after *k* steps, the finally augmented training set will enumerate to the double number of instances. Additionally, since each u_i is defined by a pair of ($x_{f\,x\,1}$, y), where the scalar *y* value is not known, the assumed human oracle is defined as a function such that H^{oracle}: $R^f \rightarrow y_{actual}$, where *f* parameter denotes the dimensionality of each dataset. The corresponding pseudocode follows here:

Incremental Active Learning Framework based on SGD(loss function, reg)	
Input:	Initially collected Labeled (L^0)/Unlabeled (U^0) subsets for pool-based scenario Define the quadruple (base-cl, UncSmetric, k, R) where loss function ≡ base-cl Annotator (H^{oracle}) Budget (B) Regularization term (reg)
Process:	Compute UncInst = round(cardinality(L^0) / k) Set iter = 0 **While** B > 0 **do** Train/Update incrementally base-cl on L^{iter} Assign through $UncS_{metric}$ confidence value to each $u_i \in U^{iter}$ Remove from U the top-UncInst instances from U^{iter} Provide them to H^{oracle} and assign its decisions to their class value B := B – UncInst iter := iter + 1
Output:	Actively trained classifier (AL_{SGD}(base-cl, reg)) built on L^k
Testing:	Measure Output's learner performance over test set for any specified classification metric

Fig. 1. Pseudocode of the proposed Incremental Active Learning framework

4 Experimental Procedure and Results

In order to verify the efficacy of the proposed AL framework, 19 binary datasets have been selected by UCI dataset. Their details are described in Table 1, along with the corresponding cardinality of initial training set (L^0) for all the selected *R*-based scenarios: 5%, 15% and 25%. Moving further, all the conducted experiments are implemented using the libact library [31] that supports AL pool-based approaches via well-known python libraries [32]. Thus, 3 different metrics have been inserted into Query Strategy of the proposed framework: Smallest Margin (sm), Least Confident (lc) and Entropy (ent), apart from Random Sampling (random) variant, which constitutes the baseline strategy of AL concept. Moreover, each Supervised approach is included into our comparisons, so as to verify both the relative improvement and the corresponding importance of the implemented algorithms per both *R*-based case and operation mode.

Table 1. Table captions should be placed above the tables.

Dataset	Instances	Features	L's cardinality for R = 5% – 15% – 25%
bands	365	20	16 – 49 – 82
breast-cancer	286	49	13 – 39 – 64
bupa	345	7	16 – 47 – 78
chess	3196	39	144 – 431 – 719
colic	368	472	17 – 50 – 83
credit-a	690	44	31 – 93 – 155
credit-g	1000	62	45 – 135 – 225
heart-statlog	270	14	12 – 36 – 61
heart	270	14	12 – 36 – 61
housevotes	232	17	10 – 31 – 52
kr-vs-kp	3196	41	144 – 431 – 719
mammographic	830	6	37 – 112 – 187
monk-2	432	7	19 – 58 – 97
pima	768	9	35 – 104 – 173
saheart	462	10	21 – 62 – 104
sick	3772	34	170 – 509 – 849
tic-tac-toe	958	28	43 – 129 – 216
vote	435	17	20 – 59 – 98
wdbc	569	31	26 – 77 – 128

Regarding the base learners that would be combined with SGD optimizer, 6 different approaches are presented here. This means that 2 different choices of base-cl parameter were made, along with all the 3 regularization terms that were referred. To be more specific, and at the same time following the notation of scikit-learn library [32], the corresponding loss functions, using their default properties, are:

- base-cl = 'log', which implements the well-known Logistic Regression learner, whose output is filtered appropriately so as to be used in classification tasks [33],
- base-cl = 'mhuber' (or 'modified huber'), which implements a smoothed hinge loss function that is equivalent to quadratically smoothed Support Vector Machine (SVM) with gamma parameter equals to 2, offering robust behavior to outliers.

To begin with, a comparison of the time response of the exploited IncL against their corresponding batch-mode variants is presented. Thus, all the 12 supervised algorithms, either incrementally updated or operating under batch-mode, are measured regarding their execution time during 10-fold cross validation (10-CV) procedure, along with two approaches that are based on the well-known Naive Bayes (NB) algorithm, so as to compare the exploited learners with algorithms that are popular for their simplicity and support also incremental update. Because of lack of space, only a sample of the produced results will be presented here. An appropriate link with the full volume of our results is provided in the end of this Section. From the depicted results, it is observed a speed-up of at least 20% for the incremental SGD-based learners, while their time performance is comparable with the MNB. Similar kind of improvement is also met into the proposed framework. A quad-core machine (Intel Core Q9300, 2.50 GHz, 8GB RAM) was used.

Table 2. Execution time of Incremental and Batch-mode supervised classifiers for 10-CV.

Dataset	SGD(mhuber, l1)		SGD(mhuber, l2)		SGD(mhuber, elnet)		Multinomial NB	
	IncL	Batch	IncL	Batch	IncL	Batch	IncL	Batch
bands	0.184	0.169	0.274	0.519	0.279	0.241	0.332	0.176
breast-cancer	0.224	0.225	0.526	0.197	0.147	0.277	0.217	0.225
bupa	0.206	0.235	0.366	0.401	0.465	0.457	0.253	0.179
chess	1.414	1.448	0.847	0.788	0.676	3.862	1.685	3.773
colic	0.409	0.306	0.221	0.182	0.418	0.486	0.223	0.936
credit-a	0.277	0.387	0.249	0.714	0.288	0.644	0.276	0.385
credit-g	0.570	1.825	0.351	1.480	1.118	0.927	0.783	0.811
heart-statlog	0.156	0.689	0.590	0.281	0.354	0.250	0.169	0.485
heart	0.141	0.152	0.149	0.244	0.221	0.285	0.234	0.217
housevotes	0.150	0.147	0.210	0.429	0.135	0.164	0.292	0.234
kr-vs-kp	0.895	1.482	0.683	1.310	2.883	3.033	1.123	1.880
mammographic	0.250	0.170	0.336	0.490	0.142	0.213	0.167	0.183
monk-2	0.126	0.206	0.139	0.316	0.156	0.164	0.158	0.156
pima	0.229	0.324	0.178	0.291	0.172	0.963	0.318	0.546
saheart	0.161	0.253	0.214	0.286	0.227	0.263	0.516	0.382
sick	2.939	2.363	1.239	1.901	2.601	1.948	1.156	1.525
tic-tac-toe	0.477	0.474	0.383	0.406	0.761	0.705	0.399	0.572
vote	0.167	0.271	0.276	0.747	0.408	0.698	0.210	0.304
wdbc	0.299	0.733	0.382	0.253	0.785	0.401	0.401	0.441
Total time (sec)	*9.148*	*11.859*	*7.613*	*11.235*	*12.236*	*15.981*	*8.912*	*13.410*

As it concerns the classification accuracy that was scored by the selected algorithms, the number of iterations has been fixed equal to 15. This value has been selected via empirical process. However, its tuning could provide better results, compromising the spent human effort and the available *B*. The next Table presents only the averaged accuracies over the 19 selected datasets, so as to compare the achieved accuracy per actively trained classifier against random strategy and the corresponding supervised variant that uses the whole dataset. The format of the next Table enables the direct comparison of IncL and batch-based approaches. The accuracy of the best performed metric per *R*-based scenario and same base-learner, independently of its operation mode, is highlighted in bold format. Only two *R*-scenarios have been included in Table 2.

It is evident that the incrementally based algorithms obtain a superior learning behavior against the conventional batch-mode operating approaches, since in all cases they outperformed the latter approaches. For providing a more detailed insight of the obtained results concerning the produced AL algorithms of the proposed framework, we notice that: in all the 90 1-vs-1 comparison between IncL and batch-based learner the former prevailed, sm metric was ranked as the best metric in 13 out of 18 cases, UncS strategy outperformed random sampling in 33 out of 54 cases, while the Supervised approaches were also outreached 22 times, regarding the incremental scenario. Keeping in mind that the proposed algorithms consume less computational resources, it seems that this kind of combination leads to more remarkable ML tools, regarding both the aspects of accuracy and time efficacy.

Table 3. Classification accuracy of Incremental and Batch-mode algorithms for 10-CV.

	R = 5%				R = 25%				Super
	lc	ent	sm	random	lc	ent	sm	random	
SGDinc(mhuber,l1)	71.36	72.98	**73.95**	73.82	**79.42**	78.71	79.32	78.10	77.89
SGDbatch(mhuber,l1)	66.90	67.41	68.87	67.49	74.26	73.33	72.36	73.44	72.12
SGDinc(mhuber,l2)	70.91	72.07	**73.24**	72.85	77.59	77.91	**78.76**	76.22	75.23
SGDbatch(mhuber,l2)	65.58	66.55	68.23	66.22	71.78	71.56	73.24	72.61	71.33
SGDinc(mhuber,elnet)	70.66	72.87	**74.12**	73.69	77.42	77.93	**78.28**	77.41	75.64
SGDbatch(mhuber,elnet)	66.81	66.88	66.99	67.05	72.13	72.10	72.01	73.84	70.82
SGDinc(log,l1)	72.47	74.02	74.62	**74.91**	78.14	79.56	79.88	**80.24**	77.34
SGDbatch(log,l1)	67.37	68.69	68.37	67.08	73.55	73.80	73.11	72.78	72.02
SGDinc(log,l2)	71.81	72.93	**74.35**	73.88	77.71	77.82	**78.52**	77.30	75.29
SGDbatch(log,l2)	67.45	66.15	66.09	67.00	72.41	71.92	72.64	72.56	71.96
SGDinc(log,elnet)	72.01	73.21	**74.97**	73.21	78.24	**78.61**	77.86	78.02	75.09
SGDbatch(log,elnet)	67.49	67.57	67.38	66.89	72.44	72.21	73.29	73.43	71.89

The statistical verification of the produced results is visualized through CD diagrams. According to this method, appropriate rankings are provided to a post-hoc test, in our case the Bonferroni-Dunn, computed by Friedman statistical test, and corresponding critical differences are computed for significance level equal to 0.05 [34]. Every algorithm that is connected via a horizontal line to another one, depicts that their learning behavior did not present significant difference.

For obtaining a more explanatory view of these comparisons, a series of violin plots has been selected to highlight the differences of the IncL and batch-mode algorithms. Through this tactic, the distribution of the scored classification accuracies is visualized, along with the average and the quartile values. In figure 3, the corresponding algorithms that use 'elnet' regularization term are presented. The complete results are provided in https://github.com/terry07/ke80537.

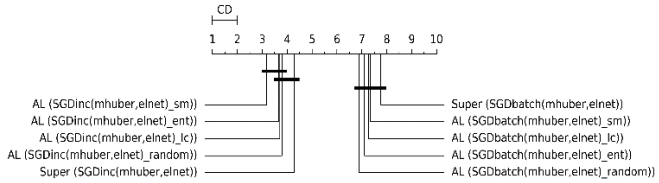

Fig. 2. A CD diagram for mhuber-based learners and 'elnet' as regularization term for R = 25%.

5 Conclusions and future work

This work constitutes a primal product of our research in involving IncL under SSL schemes so as to compensate the iterative character of the latter, by exploiting the beneficial refinement assets of the former, regarding the base learner. Our results over a wide range of binary datasets prove the remarkable classification accuracy that was achieved in case of AL concept, based on 3 amounts of labeled examples and relying on an ideal human oracle for annotation stage. The common factor over all these experiments was the use of SGD method that injects its incremental property over the linear learners that are applied. Three different regularization terms were also used, creating

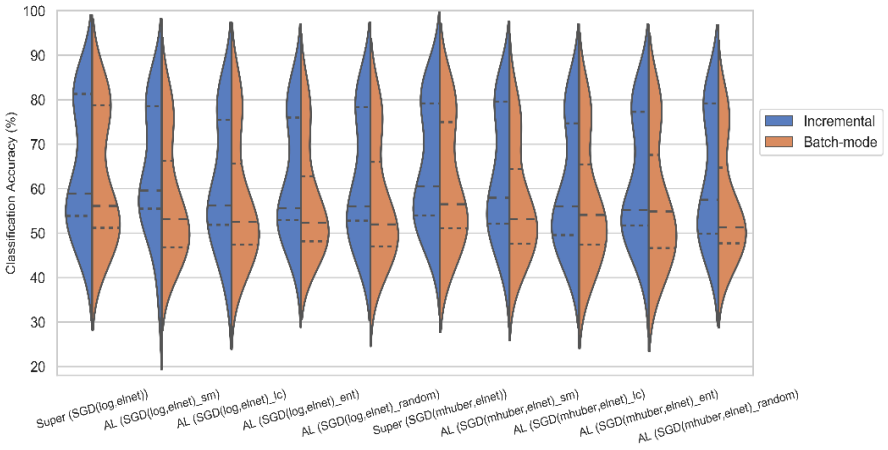

Fig. 2. A violin plot of log-based learners and 'elnet' as regularization term for R = 5%.

a series of SGD-based learners, whose learning behavior outperformed the baseline of Random Sampling strategy and the corresponding conventional batch-based methods, in the majority of the examined cases, while their performance, mainly under the Smallest Margin metric into Uncs strategy, was similar enough with their supervised rival.

The next steps are oriented towards both the examination of multiclass datasets and binary datasets that come from more specific tasks, like intrusion detections that suffers from distribution drifting [35] or text classification [36]. Furthermore, a larger variety of AL query strategies could be applied, exploiting either more sophisticated ML techniques [37] or margin-based queries that perform robustness over noisy input data [38]. Moreover, the scheme of ALBL (Active Learning By Learning) [39] could be a really promising solution, where a number of AL strategies are evaluated through a meta-learning stage. Finally, combination of SSL and AL strategies seems a powerful combination [40], reducing heavily human effort, since only a small number of iterations could be selected to ask feedback, while the incremental asset could be retained.

6 Acknowledgements

This research is implemented through the Operational Program Human Resources Development, Education and Lifelong Learning and is co-financed by the European Union (European Social Fund) and Greek national funds.

References

1. Domingos, P., Hulten, G.: Mining High-Speed Data Streams. In: KDD. p. . 71-80 (2000).
2. Pratama, M., Anavatti, S.G., Lughofer, E.: An Incremental Classifier from Data Streams. Presented at the (2014).
3. Mahmoud, M.: Semi-Supervised Keyword Spotting in Arabic Speech Using Self-Training

Ensembles. (2015).
4. Schwenker, F., Trentin, E.: Pattern classification and clustering: A review of partially supervised learning approaches. Pattern Recognit. Lett. 37, 4–14 (2014).
5. Aggarwal, C.C., Kong, X., Gu, Q., Han, J., Yu, P.S.: Active Learning: A Survey. In: Data Classification: Algorithms and Applications. pp. 571–605 (2014).
6. Zhang, Z., Cummins, N., Schuller, B.: Advanced Data Expoitation in Speech Analysis. IEEE Signal Process. Mag. 107–129 (2017).
7. Hoens, T.R., Polikar, R., Chawla, N. V: Learning from streaming data with concept drift and imbalance: an overview. Prog. {AI}. 1, 89–101 (2012).
8. Dou, J., Li, J., Qin, Q., Tu, Z.: Moving object detection based on incremental learning low rank representation and spatial constraint. Neurocomputing. 168, 382–400 (2015).
9. Bai, X., Ren, P., Zhang, H., Zhou, J.: An incremental structured part model for object recognition. Neurocomputing. 154, 189–199 (2015).
10. Tasar, O., Tarabalka, Y., Alliez, P.: Incremental Learning for Semantic Segmentation of Large-Scale Remote Sensing Data. CoRR. abs/1810.1, (2018).
11. Ristin, M., Guillaumin, M., Gall, J., Van Gool, L.: Incremental Learning of Random Forests for Large-Scale Image Classification. IEEE Trans. Pattern Anal. Mach. Intell. 38, 490–503 (2016).
12. Shin, G., Yooun, H., Shin, D., Shin, D.: Incremental learning method for cyber intelligence, surveillance, and reconnaissance in closed military network using converged IT techniques. Soft Comput. 22, 6835–6844 (2018).
13. Dou, J., Li, J., Qin, Q., Tu, Z.: Robust visual tracking based on incremental discriminative projective non-negative matrix factorization. Neurocomputing. 166, 210–228 (2015).
14. Wibisono, A., Jatmiko, W., Wisesa, H.A., Hardjono, B., Mursanto, P.: Traffic big data prediction and visualization using Fast Incremental Model Trees-Drift Detection (FIMT-DD). Knowledge-Based Syst. 93, 33–46 (2016).
15. Wang, M., Wang, C.: Learning From Adaptive Neural Dynamic Surface Control of Strict-Feedback Systems. IEEE Trans. Neural Networks Learn. Syst. 26, 1247–1259 (2015).
16. Zhang, H., Wu, P., Beck, A., Zhang, Z., Gao, X.: Adaptive incremental learning of image semantics with application to social robot. Neurocomputing. 173, 93–101 (2016).
17. Khan, S., Wollherr, D.: IBuILD: Incremental bag of Binary words for appearance based loop closure detection. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 5441–5447. IEEE (2015).
18. Saveriano, M., An, S., Lee, D.: Incremental kinesthetic teaching of end-effector and null-space motion primitives. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 3570–3575. IEEE (2015).
19. Thrun, S., Sebastian: Toward robotic cars. Commun. ACM. 53, 99 (2010).
20. Shen, Y., Yun, H., Lipton, Z.C., Kronrod, Y., Anandkumar, A.: Deep Active Learning for Named Entity Recognition. In: Blunsom, P., Bordes, A., Cho, K., Cohen, S.B., Dyer, C., Grefenstette, E., Hermann, K.M., Rimell, L., Weston, J., and Yih, S. (eds.) Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017. pp. 252–256. Association for Computational Linguistics (2017).
21. Sener, O., Savarese, S.: Active Learning for Convolutional Neural Networks: A Core-Set Approach. In: International Conference on Learning Representations (2018).
22. Liu, Y., Li, Z., Zhou, C., Jiang, Y., Sun, J., Wang, M., He, X.: Generative Adversarial Active Learning for Unsupervised Outlier Detection. CoRR. abs/1809.1, (2018).
23. Fang, M., Li, Y., Cohn, T.: Learning how to Active Learn: A Deep Reinforcement Learning Approach. In: Palmer, M., Hwa, R., and Riedel, S. (eds.) EMNLP 2017,

Copenhagen, Denmark. pp. 595–605. Association for Computational Linguistics (2017).
24. Contardo, G., Denoyer, L., Artières, T.: A Meta-Learning Approach to One-Step Active-Learning. In: Brazdil, P., Vanschoren, J., Hutter, F., and Hoos, H. (eds.) AutoML@PKDD/ECML. pp. 28–40. CEUR-WS.org (2017).
25. Krempl, G., Kottke, D., Lemaire, V.: Optimised probabilistic active learning (OPAL): For fast, non-myopic, cost-sensitive active classification. Mach. Learn. 100, 449–476 (2015).
26. Zhang, T.: Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms. In: ICML. pp. 919–926 (2004).
27. Settles, B.: Active Learning. Morgan & Claypool Publishers (2012).
28. Sharma, M., Bilgic, M.: Evidence-based uncertainty sampling for active learning. Data Min. Knowl. Discov. 31, 164–202 (2017).
29. Tsuruoka, Y., Tsujii, J., Ananiadou, S.: Stochastic Gradient Descent Training for L1-regularized Log-linear Models with Cumulative Penalty. In: ACL/IJCNLP. pp. 477–485 (2009).
30. Zou, H., Zou, H., Hastie, T.: Regularization and variable selection via the Elastic Net. J. R. Stat. Soc. Ser. B. 67, 301--320 (2005).
31. Yang, Y.-Y., Lee, S.-C., Chung, Y.-A., Wu, T.-E., Chen, S.-A., Lin, H.-T.: libact: Pool-based Active Learning in Python. (2017).
32. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Müller, A.C., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software: experiences from the scikit-learn project. In: CoRR abs/1309.0238 (2013).
33. Harrell, F.E.: Regression Modeling Strategies. Springer New York, NY (2015).
34. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. (2009).
35. Xiang, Z., Xiao, Z., Wang, D., Georges, H.M.: Incremental semi-supervised kernel construction with self-organizing incremental neural network and application in intrusion detection. J. Intell. Fuzzy Syst. 31, 815–823 (2016).
36. Lin, Y., Jiang, X., Li, Y., Zhang, J., Cai, G.: Semi-supervised collective extraction of opinion target and opinion word from online reviews based on active labeling. J. Intell. Fuzzy Syst. 33, 3949–3958 (2017).
37. Akusok, A., Eirola, E., Miche, Y., Gritsenko, A.: Advanced Query Strategies for Active Learning with Extreme Learning Machine. In: ESANN. pp. 105–110 (2017).
38. Wang, Y., Singh, A.: Noise-adaptive Margin-based Active Learning for Multi-dimensional Data. CoRR. abs/1406.5, (2014).
39. Hsu, W.-N., Lin, H.-T.: Active Learning by Learning. In: Bonet, B. and Koenig, S. (eds.) Proceedings of the Twenty-Ninth {AAAI} Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, {USA.}. pp. 2659–2665. {AAAI} Press (2015).
40. Zhao, J., Liu, N., Malov, A.: Safe semi-supervised classification algorithm combined with active learning sampling strategy. J. Intell. Fuzzy Syst. 35, 4001–4010 (2018).