Incremental learning algorithms and applications
Alexander Gepperth, Barbara Hammer

To cite this version:
Alexander Gepperth, Barbara Hammer. Incremental learning algorithms and applications. European Symposium on Artificial Neural Networks (ESANN), 2016, Bruges, Belgium. hal-01418129

HAL Id: hal-01418129
https://hal.archives-ouvertes.fr/hal-01418129
Submitted on 16 Dec 2016
Incremental learning algorithms and applications

Alexander Gepperth\textsuperscript{1}, Barbara Hammer\textsuperscript{2} \textsuperscript{*}

\textsuperscript{1} U2IS, ENSTA ParisTech
\textsuperscript{2} Bielefeld University, CITEC centre of excellence

\begin{abstract}
 Incremental learning refers to learning from streaming data, which arrive over time, with limited memory resources and, ideally, without sacrificing model accuracy. This setting fits different application scenarios where lifelong learning is relevant, e.g. due to changing environments, and it offers an elegant scheme for big data processing by means of its sequential treatment. In this contribution, we formalise the concept of incremental learning, we discuss particular challenges which arise in this setting, and we give an overview about popular approaches, its theoretical foundations, and applications which emerged in the last years.
\end{abstract}

\section{What is incremental learning?}

Machine learning methods offer particularly powerful technologies to infer structural information from given digital data; still, the majority of current applications restrict to the classical batch setting: data are given prior to training, hence meta-parameter optimisation and model selection can be based on the full data set, and training can rely on the assumption that the data and its underlying structure are static. Lifelong learning, in contrast, refers to the situation of continuous model adaptation based on a constantly arriving data stream \cite{38, 149}. This setting is present whenever systems act autonomously such as in autonomous robotics or driving \cite{156, 5, 112, 65}. Further, online learning becomes necessary in any interactive scenario where training examples are provided based on human feedback over time \cite{134}. Finally, many digital data sets, albeit static, can become so big that they are de facto dealt with as a data stream, i.e. one incremental pass over the full data set \cite{116}. Incremental learning investigates how to learn in such a streaming setting. It comes in various forms in the literature, and the use of the term is not always consistent. Therefore, first, we give a meaning to the terms online learning, incremental learning, and concept drift, giving particular attention to the supervised learning paradigm.

\subsection{Online learning methods}

In supervised learning, data $D = ((\vec{x}_1, y_1), (\vec{x}_2, y_2), (\vec{x}_3, y_3), \ldots, (\vec{x}_m, y_m))$ are available with input signals $\vec{x}_i$ and outputs $y_i$. The task is to infer a model $M \approx p(y|\vec{x})$ from such data. Machine learning algorithms are often trained in a batch mode, i.e., they use all examples $(\vec{x}_i, y_i)$ at the same time, irrespective of their (temporal) order, to perform, e.g., a model optimisation step.

\textsuperscript{*}This research/work was supported by the Cluster of Excellence Cognitive Interaction Technology ’CITEC’ (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).
Challenge 1: Online model parameter adaptation. In many application examples, data $D$ are not available priorly, but examples arrive over time, and the task is to infer a reliable model $M_t$ after every time step based on the example $(\vec{x}_t, y_t)$ and the previous model $M_{t-1}$ only. This is realised by online learning approaches, which use training samples one by one, without knowing their number in advance, to optimise their internal cost function. There is a continuum of possibilities here, ranging from fully online approaches that adapt their internal model immediately upon processing of a single sample, over so-called mini-batch techniques that accumulate a small number of samples, to batch learning approaches, which store all samples internally.

Online learning is easily achieved by stochastic optimisation techniques such as online back-propagation, but there are also extensions of the support vector machine (SVM) [164]. Prototype-based models such as vector quantisation, radial basis function networks (RBF), supervised learning vector quantisation, and self-organising maps (SOM) all naturally realise online learning schemes since they rely on a (approximate) stochastic gradient technique [140, 115, 15, 83]. Second order numeric optimisation methods and advanced optimisation schemes can be extended as well, such as variational Bayes, convex optimization, second order perceptron learning based on higher order statistics in primal or dual space, and online realisations of the quasi-Newton Broyden-Fletcher-Goldfarb-Shanno technique [114, 117, 49, 125, 62]. Stochastic optimization schemes can be developed also for non-decomposable cost function, [80]. Further, lazy learners such as k-nearest neighbour (k-NN) methods lend itself to online scenarios by their design [140]. Interestingly, online learning has already very early been accompanied by their exact mathematical investigations [162].

1.2 Incremental learning methods

Incremental learning refers to online learning strategies which work with limited memory resources. This rules out approaches which essentially work in batch mode for the inference of $M_t$ by storing all examples up to time step $t$ in memory; rather, incremental learning has to rely on a compact representation of the already observed signals, such as an efficient statistics of the data, an alternative compact memory model, or an implicit data representation in terms of the model parameters itself. At the same time, it has to provide accurate results for all relevant settings, despite its limited memory resources.

Challenge 2: Concept drift. Incremental learning shares quite a number of challenges with online learning, with memory limitations adding quite a few extras. One prominent problem consists in the fact that, when the temporal structure of data samples is taken into account, one can observe changes in data statistics that occur over time, i.e. samples $(\vec{x}_i, y_i)$ are not i.i.d. Changes in the data distribution over time are commonly referred to as concept drift [88, 157, 33, 126]. Different types of concept drift can be distinguished: changes in the input distribution $p(\vec{x})$ only, referred to as virtual concept drift or covariate shift, or changes in the underlying functionality itself $p(y|\vec{x})$, referred to as real concept drift. Further, concept drift can be gradual or abrupt. In the latter case one often uses the term concept shift. The term local concept drift characterises changes of the data statistics only in a specific region of data space [157]. A prominent example is the addition of a new, visually dissimilar object class to a
A classification problem. Real concept drift is problematic since it leads to conflicts in the classification, for example when a new but visually similar class appears in the data: this will in any event have an impact on classification performance until the model can be re-adapted accordingly.

**Challenge 3: The stability-plasticity dilemma.** In particular for noisy environments or concept drift, a second challenge consists in the question when and how to adapt the current model. A quick update enables a rapid adaptation according to new information, but old information is forgotten equally quickly. On the other hand, adaptation can be performed slowly, in which case old information is retained longer but the reactivity of the system is decreased. The dilemma behind this trade-off is usually denoted the stability-plasticity dilemma, which is a well-known constraint for artificial as well as biological learning systems [113]. Incremental learning techniques, which adapt learned models to concept drift only in those regions of the data space where concept drift actually occurs, offer a partial remedy to this problem. Many online learning methods alone, albeit dealing with limited resources, are not able to solve this dilemma since they exhibit a so-called catastrophic forgetting behaviour [108, 132, 45, 44, 103] even when the new data statistics do not invalidate the old ones.

One approach to deal with the stability-plasticity dilemma consists in the enhancement of the learning rules by explicit meta-strategies, when and how to learn. This is at the core of popular incremental models such as ART networks [77, 56], or meta-strategies to deal with concept drift such as the just-in-time classifier JIT [3], or hybrid online/offline methods [120, 43]. One major ingredient of such strategies consist in a confidence estimation of the actual model prediction, such as statistical tests, efficient surrogates, or some notion of self-evaluation [8, 43, 78]. Such techniques can be enhanced to complex incremental schemes for interactive learning or learning scaffolding [130, 84].

**Challenge 4: Adaptive model complexity and meta-parameters.** For incremental learning, model complexity must be variable, since it is impossible to estimate the model complexity in advance if the data are unknown. Depending on the occurrence of concept drift events, an increased model complexity might become necessary. On the other hand, the overall model complexity is usually bounded from above by the limitation of the available resources. This requires the intelligent reallocation of resources whenever this limit is reached. Quite a number of approaches propose intelligent adaptation methods for the model complexity such as incremental architectures [166], self-adjustment of the number of basic units in extreme learning machines [31, 177] or prototype-based models [77, 144, 98], incremental base function selection for a sufficiently powerful data representation [23], or self-adjusting cluster numbers in unsupervised learning [79]. Such strategies can be put into the more general context of self-evolving systems, see e.g. [92] for an overview. An incremental model complexity is not only mandatory whenever concept drift is observed, hence a possibly changing model complexity is present, but it can also dramatically speed-up learning in batch scenarios, since it makes often tedious model selection superfluous.

In batch learning, not only the model complexity, but also essential meta-parameters such as learning rate, degree of error, regularisation constants, etc. are determined prior to training. Often, time consuming cross-validation is used in batch learning, whereby first promising result how to automate the process
have been presented [155]. However, these are not suited for incremental learning scenarios: Concept drift turns critical meta-parameters such as the learning rate into model parameters, since their choice has to be adapted according to the (changing) data characteristics. Due to this fact, incremental techniques often rely on models with robust meta-parameters (such as ensembles), or they use meta-heuristics how to adapt these quantities during training.

**Challenge 5: Efficient memory models.** Due to their limited resources, incremental learning models have to store the information provided by the observed data in compact form. This can be done via suitable system invariants (such as the classification error for explicit drift detection models [33]), via the model parameters in implicit form (such as prototypes for distance-based models [63]), or via an explicit memory model [96, 98]. Some machine learning models offer a seamless transfer of model parameters and memory models, such as prototype- or exemplar-based models, which store the information in the form of typical examples [63]. Explicit memory models can rely on a finite window of characteristic training examples, or represent the memory in the form of a parametric model. For both settings, a careful design of the memory adaptation is crucial since it directly mirrors the stability-plasticity dilemma [96, 98].

**Challenge 6: Model benchmarking.** There exist two fundamentally different possibilities to assess the performance of incremental learning algorithms: (1) **Incremental -vs- non-incremental:** In particular in the absence of concept drift, the aim of learning consists in the inference of the stationary distribution \( p(y|\vec{x}) \) for typical data characterised by \( p(\vec{x}) \). This setting occurs e.g. whenever incremental algorithms are used for big data sets, where they compete with often parallelised batch algorithms. In such settings, the method of choice evaluate the classification accuracy of the final model \( M_t \) on a test set, or within a cross-validation. While incremental learning should attain results in the same range as batch variants, one must take into account that they deal with restricted knowledge due to their streaming data access. It has been shown, as an example, that incremental clustering algorithms cannot reach the same accuracy as batch versions if restricted in terms of their resources [2].

(2) **Incremental -vs- incremental:** When facing concept drift, different cost functions can be of interest. Virtual concept drift aims for the inference of a stationary model \( p(y|\vec{x}) \) with drifting probability \( p(\vec{x}) \) of the inputs. In such settings, the robustness of the model when evaluated on test data which follow a possibly skewed distribution is of interest. Such settings can easily be generated e.g. by enforcing imbalanced label distributions for test and training data [73]. Whenever real confidence drift is present, the online behaviour of the classification error \( \| M_t(\vec{x}_{t+1}) - y_{t+1} \| \) for the next data point is usually the method of choice; thereby, a simple average of these errors can be accompanied by a detailed inspection of the overall shape of the online error, since it provides insight into the rates of convergence e.g. for abrupt concept drift.

(3) **Formal guarantees on the generalisation behaviour:** Since many classical algorithms such as the simple perceptron or large margin methods have been proposed as online algorithms, there exists an extensive body of work investigating their learning behaviour, convergence speed, and generalisation ability, classically relying on the assumption of data being i.i.d. [162]. Some results weaken the i.i.d. assumption e.g. requiring only interchangeability [146]. Re-
cently, popular settings such as learning a (generalised) linear regression could be accompanied by convergence guarantees for arbitrary distributions \( p(\tilde{x}) \) by taking a game theoretic point of view: in such settings, classifier \( M_t \) and training example \( \tilde{x}_{t+1} \) can be taken in an adversarial manner, still allowing fast convergence rates in relevant situations [158, 131, 87, 151]. The approach [117] even provides first theoretical results for real context drift, i.e. not only the input distribution, but also the conditional distribution \( p(y|x) \) can follow mild changes.

2 Incremental learning models

Incremental learning comes in various forms in the literature, and the use of the term is not always consistent; for some settings, as an example, a memory limitation cannot be guaranteed, or models are designed for stationary distributions only. We will give an overview over popular models in this context. Thereby, we will mostly focus on supervised methods due to its popularity. Online or incremental learning techniques have also been developed for alternative tasks such as clustering [109, 91], dimensionality reduction [25, 12, 24, 6, 123, 93], feature selection and data representation [179, 59, 42, 173, 27, 72], reinforcement learning [11, 60], mining and inference [54, 129].

Explicit treatment of concept drift. Dealing with concept drift at execution time constitutes a challenging task [88, 157, 33, 126]. There exist different techniques to address concept drift, depending on its type. Mere concept shift is often addressed by so-called passive methods, i.e. learning technologies which smoothly adapt model parameters such that the current distribution is reliably represented by the model. Rapid concept changes, however, often require active methods, which detect concept drift and react accordingly.

Virtual concept drift, which concerns the input distribution only, can easily occur e.g. due to highly imbalanced classes over time. One popular state-of-the-art technology accounts for this fact by so-called importance weighting, i.e. strategies which explicitly or implicitly reweight the observed samples such that a greater robustness is achieved [81, 10, 73]. Alternatively, concept shift can have its reason in novelty within the data or even new classes. Such settings can naturally be incorporated into local models provided they offer an adaptive model complexity [144, 43, 56, 133, 100].

Real concept drift can be detected by its effect on characteristic features of the model such as the classification accuracy. Such quantitative features can be accompanied by statistical tests which can judge the significance of their chance, hence concept drift. Tests can rely on well-known statistics such as the Hoeffding bound [48], or alternatively on suitable distances such as the Hellinger distance, which can measure the characteristics of value distributions of such characteristic features. When integrated into robust classifiers such as ensemble techniques, models which can simultaneously deal with different types of drift result [16].

Support vector machines and generalised linear models. Several incremental SVM models exist [164]. Some rely on heuristics, like retraining a model with all support vectors plus a new “incremental” batch of data [35, 152], but without theoretical guarantees. Other incorporate modification of the SVM cost function to facilitate incrementality [141] and also possibly control complexity.
their resources are not strictly limited. As an alternative, \textit{adiabatically} SVM training has been proposed, i.e., presenting one example at a time while maintaining the relevant optimality conditions on all previously seen examples. However, this requires all previously seen samples need to be stored, although the approach can considerably simplify SVM training. Ensemble learning algorithms based on SVM [127, 164] achieve incremental learning by training new classifiers for new batches of data, and combining all existing classifiers only for decision making. Another hybrid scheme combines a SVM classifier with a prototype-based data representation, whereby the latter can be designed as an online model based on which training examples for SVM can be generated [169]. Alternatively, SVMs can directly be trained in primal space, where online learning is immediate [22]. Online versions have also been proposed for more complex generalised linear models such as Gaussian Process regression [53, 110], whereby none of these models can yet easily deal with concept drift.

\textbf{Connectionist models.} As the problem of catastrophic forgetting was first remarked for multilayer perceptrons (MLP) [108, 132], it is hardly surprising that there exists significant work how to avoid it in connectionist systems. Initial consensus traced catastrophic forgetting back to their \textit{distributed} information representation [46]. Indeed, \textit{localist} connectionist models such as radial basis function (RBF) networks can work reliably in incremental settings [133, 100], whereby care has to be taken to guarantee their generalisation performance [147]. Both capabilities are combined in \textit{semi-distributed representations}. A number of algorithmic modifications of the MLP model has been proposed, such as sparsification [45], orthogonalization of internal node weights [47, 119], reduction of representational overlap while training [85], or specific regularisation [55]. These are successful in mitigating but not eliminating catastrophic forgetting [147]. Recently, there has been an increased interest in extreme learning machines (ELM), which combine a random mapping with a trained linear readout. Due to their simple training, incremental variants can easily be formulated, whereby their reservoir naturally represents rich potential concepts [178, 31, 159, 61].

Furthermore, there exist attempts to modify the system design of MLPs [86, 150] which are more in the line of generative learning: they incorporate novelty detection and use different representational resources for new samples. Elaborate connectionist models feature different memory subsystems for long- and short-term learning [139, 7], as well as explicit replay and re-learning of previous samples to alleviate forgetting [135]. These approaches reduce the problem of catastrophic forgetting at the price of vastly more complex model. Contrarily to other modern approaches, inspiration is taken primarily from biology and thus its solid mathematical understanding is yet lacking.

\textbf{Explicit partitioning approaches.} Many modern incremental learners rely on a local partitioning of the input space, and a separate classification/regression model for each partition [160, 121, 148, 18, 21]. The manner of performing this partitioning is very diverse, ranging from kd-trees [21] to genetic algorithms [18] and adaptive Gaussian receptive fields [160]. Equally, the choice of local models varies between linear models [160], Gaussian mixture regression [21] or Gaussian Processes [121]. For high-dimensional problems such as occur in perception, the partitioning of the input space constitutes the bottleneck as concerns memory consumption. Covariance matrices as used in [160], for example, are quadratic
in the number of input dimensions, hence prohibitive for high dimensional data.

Decision trees partially alleviate this problem insofar as they cut along one dimension only, disregarding feature correlations. Quite a number of incremental tree builders have been proposed for classification or regression [41, 142, 52], with a particular focus on when to split, how to avoid overly large trees while ensuring incremental growth, and how to reliably deal with imbalanced classes [66, 26, 102]. Interestingly, there do exist tree classifiers which result is entirely invariant to the ordering of the training data, but at the price of unlimited resources [90].

**Ensemble methods.** Ensemble methods combine a collection of different models by a suitable weighting strategy. As such, they are ideally suited to implicitly represent even partially contradictory concepts in parallel and mediate the current output according to the observed data statistics at hand. Ensemble methods have proved particularly useful when dealing with concept drift, with a few popular models ranging from incremental random forests [105], ensembles of bipartite graph classifiers [13], up to advanced weighting schemes suitable for different types of concept drift and recurring concepts [172, 111, 95, 172, 32, 39].

**Prototype-based methods.** Prototype-based machine learning has its counterpart in cognitive psychology [137] which hypothesises that semantic categories in the human mind are represented by specific examples for these categories. In machine learning approaches, a class is represented by a number of representatives, and class membership is defined based on the distance of the data from these prototypes. For high dimensional data, adaptive low-rank metric learning schemes can dramatically improve classification accuracy and efficiency [17, 145]. Prototype-based methods are a natural continuation of the work on localist or semi-distributed representations in early connectionist models, and thus share many properties. They have the advantage of an easily adaptive model complexity. One disadvantage is that the number of prototypes can become large whenever complex class boundaries are present.

Prototype-based models are closely connected to the non-parametric k-NN classifier (all training points act as prototypes) and the RBF model [140]. A popular supervised method is given by learning vector quantisation (LVQ) and recent variants which can be substantiated by a cost function [15]. A number of incremental variants and methods capable of dealing with concept drift have been proposed, such as dynamic prototype inversion / deletion schemes [98, 144], or techniques with fixed model complexity, but intelligent source redistribution strategies [50]. Similar unsupervised incremental models exist [63, 19, 176].

**Insights into biological incremental learning.** As biological incremental learning has reached a high degree of perfection, biological paradigms can provide inspiration how to set up artificial incremental systems. There is evidence that sensory representations in the neocortex are prototype-based, whereby neurons are topologically arranged by similarity [153, 94, 138, 40]. Learning acts on these representations in a task-specific way insofar as the density of neurons is correlated to sensory regions which require finer discrimination [128], i.e., where more errors occur. Here, learning is conceivably enhanced through acetylcholine release in case of task failures [163, 70]. Learning respects the topological layout by changing only a small subset of neural selectivities [136] at each learning event, corresponding to regions around the best matching unit [40].
Beyond the single-neuron level, there is a large body of literature investigating the roles of the hippocampal and neocortical areas of the brain in learning at the architectural level. Generally speaking, the hippocampus employs a rapid learning rate with separated representations whereas the neocortex learns slowly, building overlapping representations of the learned task [122]. A well-established model of the interplay between the hippocampus and the neocortex suggests that recent memories are first stored in the hippocampal system and played back to the neocortex over time [107]. This accommodates the execution of new tasks that have not been recently performed as well as the transfer of new task representations from the hippocampus (short-term memory) to the neocortical areas (long-term memory) through slow synaptic changes, i.e. it provides an architecture which is capable of facing the stability-plasticity dilemma.

3 Applications

We would like to conclude this overview by a glimpse on typical application scenarios where incremental learning plays a major role.

Data analytics and big data processing. There is an increasing interest in single-pass limited-memory models which enable a treatment of big data within a streaming setting [64]. Their aim is to reach the capability of offline techniques, and conditions are less strict as concerns e.g. the presence of concept drift. Recent approaches extend, for example, extreme learning machines in this way [168]. Domains, where this approach is taken, include image processing [34, 97], data visualization [106], and processing of networked data [29].

Robotics. Autonomous robotics and human-machine-interaction are inherently incremental, since they are open-ended, and data arrive as a stream of signals with possibly strong drift. Incremental learning paradigms have been designed in the realm of autonomous control [161], service robotics [5], computer vision [175], self-localisation [82], or interactive kinesthetic teaching [143, 51]. Further, the domain of autonomous driving is gaining enormous speed [156, 118, 4], with enacted autonomous vehicle legislation in already eight states in the US (Dec. 2015). Another emerging area, caused by ubiquitous sensors within smart phones, addresses activity recognition and modeling [69, 1, 89, 74, 99, 68].

Image processing. Image and video data are often gathered in a streaming fashion, lending itself to incremental learning. Typical problems in this context range from object recognition [36, 9, 98], image segmentation [36, 71], and image representation [30, 165], up to video surveillance, person identification, and visual tracking [154, 28, 167, 104, 37, 134, 174, 101].

Automated annotation. One important process consists in the automated annotation or tagging of digital data. This requires incremental learning approaches as soon as data arrive over time; example systems are presented in the approaches [75, 14, 20] for video and speech tagging.

Outlier detection. Automated surveillance of technical systems equipped with sensors constitutes an important task in different domains, starting from process monitoring [67] fault diagnosis in technical systems [171, 76, 170], up to cyber-security [124]. Typically, a strong drift is present in such settings, hence there is a demand for advanced incremental learning techniques.
References


