Journal of Artificial Intelligence Research 68 (2020) 159-224 Submitted 03/2019; published 05/2020

Towards Knowledgeable Supervised Lifelong Learning Systems

Diana Benavides-Prado DBEN652@AUCKLANDUNI.AC.NZ
Yun Sing Koh YKOH@CS.AUCKLAND.AC.NZ
Patricia Riddle PAT@CS.AUCKLAND.AC.NZ
School of Computer Science
The University of Auckland, New Zealand

Abstract

Learning a sequence of tasks is a long-standing challenge in machine learning. This setting applies to learning systems that observe examples of a range of tasks at different points in time. A learning system should become more knowledgeable as more related tasks are learned. Although the problem of learning sequentially was acknowledged for the first time decades ago, the research in this area has been rather limited. Research in transfer learning, multitask learning, metalearning and deep learning has studied some challenges of these kinds of systems. Recent research in lifelong machine learning and continual learning has revived interest in this problem. We propose Proficiente, a full framework for long-term learning systems. Proficiente relies on knowledge transferred between hypotheses learned with Support Vector Machines. The first component of the framework is focused on *transferring forward* selectively from a set of existing hypotheses or functions representing knowledge acquired during previous tasks to a new target task. A second component of Proficiente is focused on *transferring backward*, a novel ability of long-term learning systems that aim to exploit knowledge derived from recent tasks to encourage refinement of existing knowledge. We propose a method that transfers selectively from a task learned recently to existing hypotheses representing previous tasks. The method encourages retention of existing knowledge whilst refining. We analyse the theoretical properties of the proposed framework. Proficiente is accompanied by an agnostic metric that can be used to determine if a long-term learning system is becoming more knowledgeable. We evaluate Proficiente in both synthetic and real-world datasets, and demonstrate scenarios where knowledgeable supervised learning systems can be achieved by means of transfer.

1. Introduction

Machine learning is becoming increasingly popular. A wide range of applications and decision making processes are being supported by this technology. The learning ability of a machine learning algorithm is typically limited by the examples used for training. The typical setting in machine learning is to learn different tasks in isolation. However, learning systems such as biological systems operate different. For example, in human learning systems typically old knowledge (*e.g.* how to ride a bike) is used whenever faced with a new task (*e.g.* learning to drive a car). After new knowledge is gathered, existing related knowledge is usually refined. As a result, humans become more knowledgeable as they learn more related concepts. Human learning as a reference for the design of machine learning systems has been studied for decades (Singley & Anderson, 1987).

As an example applicable to machines or algorithms that learn, consider the problem of learning to recognise scans of the human brain for classifying patients into diagnosis categories. Learning about *brain injury*, for example, is limited by the training data. With no additional information, a hypothesis that distinguishes this class from others will have suboptimal performance on unseen cases. If a related class such as *brain disorder* is learned later on, sharing knowledge across these classes may be beneficial for both. Knowledge on existing categories can evolve as new classes are learned supported by existing knowledge. This paper studies these kinds of problems in the supervised learning setting.

Transfer learning, domain adaptation, multitask learning, metalearning and continual learning research have tackled some of the challenges of machine learning systems that learn a sequence of supervised tasks (Pan & Yang, 2010; Weiss, Khoshgoftaar, & Wang, 2016; Caruana, 1998; Brazdil, Carrier, Soares, & Vilalta, 2008; Lemke, Budka, & Gabrys, 2015; Parisi, Kemker, Part, Kanan, & Wermter, 2018). Albeit effective for a variety of problems, existing research in these fields has focused on improving the performance of new tasks. In transfer learning and domain adaptation, usually a target task is supported by data or features transferred from a single source task. Multitask learning has studied the problem of improving the performance of a set of tasks learned in parallel while sharing knowledge. Metalearning has aimed to maintain a meta-layer of knowledge that describes learning experiences of a system, which can be exploited later for learning of new tasks. Continual learning has investigated the problem of adapting a single deep neural network to the challenge of learning a sequence of tasks whilst avoiding forgetting existing knowledge.

Hypothesis transfer learning is an alternative that exploits previous models or hypotheses for knowledge transfer. In supervised learning, a hypothesis constitutes the final result of learning, and therefore is knowledge about a category that a learning system uses for classification, prediction or for future learning tasks. An example of a hypothesis is a function f of the form $Y = \mathbf{w}x + b$, that can be used to classify an example x from a sample X represented in a feature space $\mathcal{X}$ into a sample of classes or labels Y from the class or label space $\mathcal{Y}$, using a learned set of weights $\mathbf{w}$ and a bias term b. Support Vector Machines (SVM) has been one of the recent avenues of research in hypothesis transfer learning (Yang, Yan, & Hauptmann, 2007; Tommasi, Orabona, & Caputo, 2014; Kuzborskij, Orabona, & Caputo, 2015; Oneto, Ghio, Ridella, & Anguita, 2015; Mozafari & Jamzad, 2016; Wang & Hebert, 2016). SVM is a theoretically grounded technique used for both classification and regression. Three aspects contribute to the choice of this technique for knowledge transfer from source hypotheses: first, the SVM hypothesis representation as a set of support vectors encourages transfer and selective transfer from multiple sources. Since these support vectors are a subset of the training examples, these can be directly used for transfer purposes. Other techniques, such as for example neural networks, may have to generate training examples if these are required for transfer (Atkinson, McCane, Szymanski, & Robins, 2018a). Second, a variety of research on the properties of statistical learning theory has been defined as variants of the original SVM problem, *e.g.* ν-SVM (Schölkopf, Smola, Williamson, & Bartlett, 2000) and learning with privileged information (Vapnik & Vashist, 2009), which raises the potential to combine these variants to simultaneously achieve different properties of learning. Third, a variety of studies have demonstrated the potential use of additional information to improve learning of an SVM hypothesis (Schölkopf, Burges, & Vapnik, 1996; Decoste & Schölkopf, 2002; Oneto et al., 2015; Vapnik & Izmailov, 2015). A recent interest has also arisen for supervised lifelong learning research using SVM (Fei, Wang, & Liu, 2016). We aim to exploit the strengths of SVM for our purpose of achieving knowledgeable supervised lifelong learning systems. An introduction to SVM and preliminary concepts useful for this paper are presented in Section 3.

Lifelong machine learning has gained increasing attention in the last few years. Acknowledged for the first time more than two decades ago (Mitchell & Thrun, 1993; Thrun, 1996), lifelong learning has been defined as a continuous learning process that relies on the execution of a sequence of related learning tasks (Chen & Liu, 2016). Domains such as object recognition, text classification and sentiment categorization, where examples from new classes are observed at different points in time, are candidate application areas for supervised lifelong machine learning methods. A remarkable characteristic of lifelong learning systems is their ability to become more knowledgeable and perform better while more related tasks are observed. Nevertheless, the problem of improving the performance of a learning system as a whole remains a challenge (Chen & Liu, 2015, 2016), and only a few supervised lifelong learning alternatives have been proposed recently (Silver, Yang, & Li, 2013; Ruvolo & Eaton, 2013b; Chen, Ma, & Liu, 2015; Fei et al., 2016). We discuss existing research in lifelong machine learning and related areas in Section 2.

In this paper we propose **Proficiente**, a novel framework that encourages sequential learning using bi-directional and selective transfer of knowledge between SVM tasks. We aim to tackle the properties of lifelong machine learning systems identified by Chen and Liu (2016): 1) to learn new tasks better, supported by existing knowledge, 2) to perform continuous learning, by refining existing knowledge, 3) to store knowledge continuously and incrementally in a knowledge base. **Proficiente** is accompanied by a knowledge base that stores knowledge in the representational form described by Silver and Poirier (2007), where only the final result of learning is stored in the long-term. An introduction to **Proficiente** is provided in Section 4. The proposed scheme is depicted in Figure 1.

Proficiente operates in two stages: **transfer forward**, which is focused on learning a new hypothesis on a target task, and **transfer backward**, which is focused on refining existing hypotheses using knowledge collected while learning the target task. These stages are closely interconnected. The first stage relies on transferring selected fragments of knowledge from a set of source SVM hypotheses to a new related SVM task. We have proposed **AccGenSVM**, a method that identifies a subset of source hypotheses to use for transfer (Benavides-Prado, Koh, & Riddle, 2017). This method and the theoretical properties of selective transfer forward are described in Section 5. **AccGenSVM** uses selected coefficients learned for source support vectors to upper-bound coefficients on target examples which are closely related to the source support vectors. As a result, some training examples on the target task obtain larger upper-bounds. These examples make a larger contribution to the SVM hypothesis to be learned. A remarkable property of this transfer forward approach is that it only needs to access source hypotheses, without relying on source data. Therefore, the method loosens the limitation of availability of knowledge for transfer identified by Chen and Liu (2016).

Transferring backward represents a novel approach to transfer, for which research has been rather limited. The aim of transferring backward is to refine knowledge about existing classes or categories, by updating their corresponding hypotheses using knowledge collected after these hypotheses were learned. This refinement is expected to encourage better performance of these hypotheses on unobserved samples. Therefore, the learning system as a whole could potentially achieve increasing performance as more tasks are learned. This fundamental ability of lifelong machine learning systems (Silver & Poirier, 2007; Silver et al., 2013) has received very little attention. A first approximation to transferring backward explicitly was presented by us (Benavides-Prado, Koh, & Riddle, 2019). Research such as ELLA (Ruvolo & Eaton, 2013b) has approximated this problem implicitly in the context of multitask learning. Cumulative Learning, or CL (Fei et al., 2016), approximated this problem explicitly although without transfer. Recent research in continual learning has studied a similar problem in the context of avoiding *catastrophic forgetting* of existing knowledge about previous tasks (Parisi et al., 2018; Atkinson et al., 2018a; Atkinson, McCane, Szymanski, & Robins, 2018b). Continual learning has revived the interest of studying the problem of *catastrophic forgetting* which was previously studied in the context of knowledge consolidation in neural networks (Robins, 1995, 1996), some of which work also pointed to the possibility of improving knowledge of previous tasks as a result of learning new tasks (Silver, Mason, & Eljabu, 2015).

For transferring backward we propose Hypothesis Refinement with SVM, **HRSVM**, a method that aims to refine knowledge of previous tasks whenever a new task is observed. As a result of the transfer process performed by **AccGenSVM**, pairs of related source support vectors and target support vectors learned on the target hypothesis can be identified. These pairs denote subspaces of shared knowledge between the corresponding target and source hypotheses, which were originally used to aid learning of the target task. Moreover, this is additional knowledge that has become available after learning the source hypotheses, which can be potentially useful for refining these sources. **HRSVM** exploits these pairs by learning representations in the form of local functions that use source and target support vectors as training examples. Then, **HRSVM** solves a modified SVM classification problem that learns a refined version of a source hypothesis. This classification problem considers both the space of the support vectors on the source hypothesis and the local functions learned.

A fundamental aspect for lifelong learning systems in the long-term is their ability to retain existing knowledge (Silver & Poirier, 2007; Silver et al., 2013). Furthermore, the ability to learn a large and unknown number of tasks is also desired for long-term learning. However, these properties pose a challenge: the larger the number of tasks the more difficult it is to control that previous knowledge is not forgotten or corrupted (Parisi et al., 2018). This trade-off is also observed in biological systems, although only in some circumstances for humans (Bremner, Lewkowicz, & Spence, 2012). **HRSVM** aims to retain knowledge of existing hypotheses while these hypotheses are refined by solving a modified ν-SVM problem (Schölkopf et al., 2000). ν-SVM is an SVM variant that controls retention of training examples as support vectors whilst minimising the training error. Details of **HRSVM** are provided in Section 6. We also analyse the theoretical properties of **HRSVM**, and discuss the proposed transfer backward setting in the presence of related tasks, unrelated tasks, and tasks learned sequentially. In Section 7 we present experimental results of the proposed framework and counterparts in three real-world datasets and two synthetic datasets generated for supervised lifelong learning.

We accompany **Proficiente** with Cumulative Gain of a Lifelong Learner, **CGLL**, a test to measure the performance of systems that learn a sequence of tasks. Although

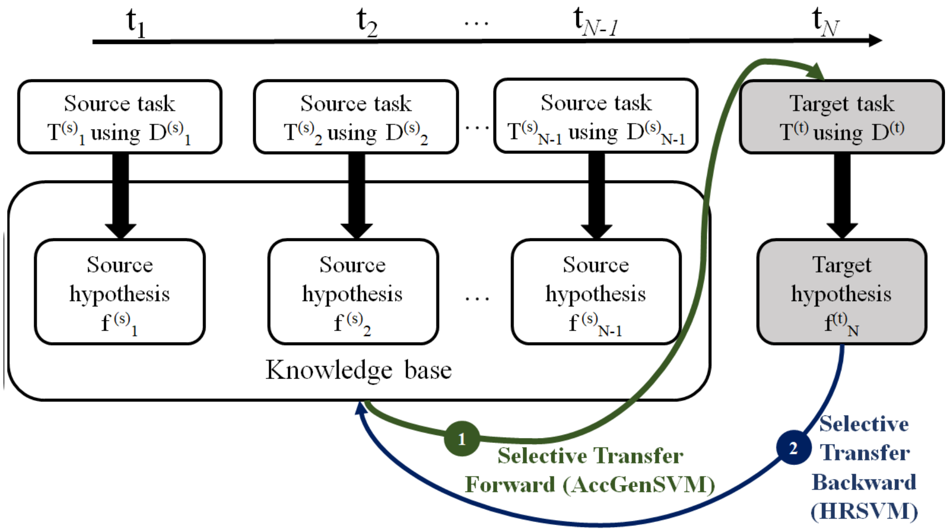


Figure 1: Our proposed scheme for learning in the long-term. At a given timestamp t, a classification task T is to be learned using a set of training examples D. Hypotheses learned on these tasks are stored in a knowledge base once learned. Hypotheses learned previously are named source hypotheses, and are denoted as $f^{(s)}$. The hypothesis to be learned in the current timestamp is named the target hypothesis, and is denoted as $f^{(t)}$. Selective transfer occurs forward to learn a target hypothesis and backward to refine existing hypotheses.

fundamental, the design of metrics for determining the performance of sequential learning systems has only been recently explored (Li & Yang, 2015; Díaz-Rodríguez, Lomonaco, Filliat, & Maltoni, 2018). Most studies in lifelong or continual learning systems have relied on classic metrics used for single-task learners. **CGLL** is described in Section 8.

This paper is organised as follows. In Section 2 we summarise and discuss previous research that has studied the problem of learning a sequence of tasks. In Section 3 we explain preliminary concepts and notation which is useful for the rest of the paper. In Section 4 we present an overview of **Proficiente**. In Section 5 we present **AccGenSVM** (Benavides-Prado et al., 2017), a method that transfers selectively from a set of SVM hypotheses to an SVM target task. We first describe previous research that has inspired the proposed method, and then formulate the problem of **AccGenSVM** and its theoretical properties. In Section 6 we present **HRSVM**, a method that refines existing hypotheses learned on previous tasks by exploiting knowledge acquired during a recent target task. We first contextualize this problem. We then discuss previous research that has tackled this problem implicitly or explicitly. We formulate the problem of **HRSVM** and describe its theoretical properties. We provide theoretical analyses of the effects of our proposed method in scenarios of related tasks, unrelated tasks and tasks learned sequentially. In Section 7 we provide experimental results of **HRSVM** together with **AccGenSVM** and counterpart methods in two synthetic datasets generated for classification tasks in lifelong learning. We also present results for three real-world datasets. We also provide experimental results using an abstract example introduced as part of **HRSVM**. In Section 8 we propose **CGLL**, a metric to determine how knowledgeable a supervised lifelong learning system is becoming as it learns more tasks. We present results of this metric and a counterpart in two synthetic datasets and three real-world datasets. Finally, in Section 9 we provide final remarks and point to future research derived from this research.

2. Previous Research on Supervised Lifelong Learning

Transfer learning has been an increasingly active research area over the last few decades. The aim of transfer learning is to aid learning of new tasks by exploiting related previous tasks or related domains, especially when target tasks suffer from scarce labelled data. Transfer across tasks or domains is challenging since generally the distributions of the source and target data are different. The goal is to learn target tasks faster or with better performance compared to learning these tasks from scratch using their training examples only. Pan and Yang (2010) distinguished three settings of transfer learning: 1) the inductive transfer learning setting where the source and target tasks are different but related, 2) the transductive transfer learning setting where the source and target domains are different but related, 3) the unsupervised learning setting. The authors also distinguished transfer learning solutions depending on the kind of information or knowledge that is transferred: 1) data transfer, 2) feature-representation transfer, 3) parameter transfer, and 4) relational knowledge transfer. More recently, Weiss et al. (2016) surveyed transfer learning methods by classifying existing research in three categories: methods for homogeneous features in the source and target tasks, methods for heterogeneous features between the source and target tasks and methods that study the problem of negative transfer.

Research surveyed by Pan and Yang (2010) and by Weiss et al. (2016) denote a remarkable characteristic of transfer learning methods: these have been usually focused on transferring from a single source task to a single target task, in a single direction. Therefore, the classic setting of transfer learning does not consider transfer from a variety of sources, nor learning several target tasks in sequence whilst encouraging transfer. Furthermore, most research has focused on transferring data or feature representations, two approaches that require source data to be available for future transfer. This fact poses a limitation for long-term learning systems that rely on knowledge transfer.

Domain adaptation is an application of transfer learning that aims to adapt data from a source domain to learn a target task on a target domain. This adaptation is typically performed at the feature representation level. Similar to transfer learning, the main challenge is to perform transfer in the presence of shifted distributions between the source and the target domains, especially when the target task suffers from a scarcity of labelled data. In supervised learning, the most recognised domain adaptation approach proposed to train a classifier using augmented features from a source and a target domain (Daumé III, 2009). Domain adaptation solutions proposed afterwards aimed to transform, correct or learn common feature representations between one or several source domains and a target domain (Duan, Tsang, Xu, & Chua, 2009; Saenko, Kulis, Fritz, & Darrell, 2010; Gong, Shi, Sha, & Grauman, 2012; Hoffman, Kulis, Darrell, & Saenko, 2012). Similar to most research in transfer learning, domain adaptation research has considered adaptation from a single source domain. The source data needs to be available for adaptation.

Hypothesis transfer learning is an alternative that considers transfer from previous models or hypotheses to aid learning of a target task. The key advantage of this kind of transfer is that data from source tasks does not need to be available, since transfer of knowledge is performed directly from previous hypotheses. This characteristic makes hypothesis transfer learning applicable to problems where the storage of previous data is limited or the privacy of the data needs to be guaranteed. Transfer is commonly performed from a set of previous models or hypotheses, rather than from a single source. A number of theoretical (Kuzborskij & Orabona, 2013; Kuzborskij, 2018), experimental (Yang et al., 2007; Aytar & Zisserman, 2011; Tommasi et al., 2014; Kuzborskij et al., 2015; Oneto et al., 2015; Mozafari & Jamzad, 2016; Wang & Hebert, 2016), and application-specific methods (Valerio, Passarella, & Conti, 2016) have been proposed. Transferring from previous hypotheses rather than from previous data as in classic transfer learning provides a potential for long-term learning systems, since these hypotheses are the only knowledge that needs to be stored to aid learning of future tasks.

Deep transfer learning is a more recent approach to transfer learning which can be classified into the parameter-based transfer learning category proposed by Pan and Yang (2010). The common approach for transfer learning in supervised deep neural networks has been to initialize the network of a target task with weights learned for a network during a previous task (Yosinski, Clune, Bengio, & Lipson, 2014). Other studies have extended deep transfer learning to domain adaptation problems, where a network architecture is built in such a way that domain discrepancies between the source and target domains are reduced, and therefore transfer can be achieved (Long, Cao, Wang, & Jordan, 2015). Similar to classic transfer learning, transfer is limited to a single source and a target task. Continual learning, which is discussed later in this section, is a variant of this approach that studies the problem of learning a sequence of tasks whilst updating a single network.

Multitask learning is an inductive transfer mechanism that explores the problem of learning multiple tasks in parallel (Caruana, 1998). The main purpose is for each task to benefit from data or knowledge from other tasks while these are learned jointly. This is achieved by learning a shared representation of these tasks which is typically represented as a set of shared parameters. The hypothesis for each task is therefore composed of both this shared set and a specific set of parameters learned for the corresponding task. This is expected to provide better performance compared to learning these tasks in isolation. A recent survey (Ruder, 2017) shows that the idea of multitask learning has been implemented in the context of deep neural networks as early as 2010. The main assumption in multitask learning is that examples from all tasks can be observed at once. However, this assumption does not hold true for long-term learning systems that learn sequentially.

Metalearning has investigated the problem of accumulating and using experience over several applications of a learning system (Brazdil et al., 2008). The common approach is to maintain a meta-layer that describes knowledge collected during previous tasks. This meta-layer could be exploited during future learning tasks. Lemke et al. (2015) described several settings of metalearning. The inductive transfer setting has gained increased attention recently, for the problems of one-shot and few-shot learning. In these kinds of problems, the aim is to learn a very generalizable model from a task with a large set of training examples, which can be applicable to future tasks where the number of training examples is very limited. The model should be learned over a domain which is related to the future tasks of interest. Examples of these settings are metalearning methods to learn a highly generalizable set of parameters that can be easily adapted for new tasks (Santoro, Bartunov, Botvinick, Wierstra, & Lillicrap, 2016; Finn, Abbeel, & Levine, 2017), Bayesian-based

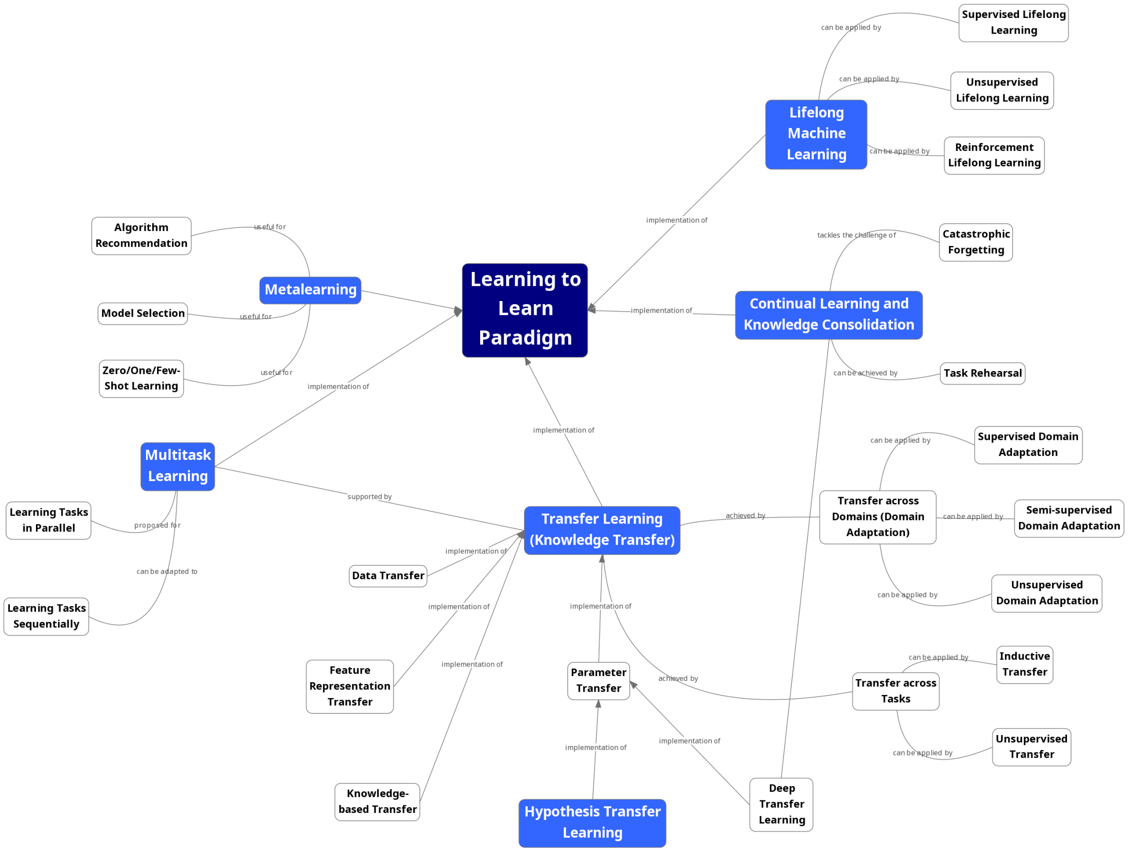


Figure 2: Fields that study the problem of learning a sequence of tasks, and related challenges.

methods (Amit & Meir, 2018; Yoon, Kim, Dia, Kim, Bengio, & Ahn, 2018), and methods based on finding low-dimensional latent spaces of features that are highly generalizable (Rusu, Rao, Sygnowski, Vinyals, Pascanu, Osindero, & Hadsell, 2018).

Lifelong machine learning is an area of increasing attention. The challenge of learning systems that work over the long-term was acknowledged for the first time more than two decades ago (Thrun, 1996). Several approaches from transfer, multitask and lifelong learning were categorised as part of the learning to learn paradigm (Thrun & Pratt, 1998). Learning to learn was defined as the ability of a learning system to improve performance as more tasks are observed and learned. This improvement should also depend on the number of tasks. The larger the number of tasks, the larger the improvement. Silver (2000) studied lifelong machine learning systems based on neural networks. More than a decade later, Silver, Yang and Li (2013) described existing research in lifelong machine learning, for supervised, unsupervised and reinforcement learning problems. The authors described lifelong learning systems that retain knowledge collected during these tasks, and which use that knowledge to learn new tasks more efficiently and effectively. More recently, Chen and Liu (2016) revived interest in lifelong machine learning systems and defined three core properties of these kinds of systems: 1) learning new tasks using knowledge from previous tasks, 2) learning continuously and incrementally, 2) retaining knowledge in a knowledge base. The authors surveyed lifelong machine learning research in the supervised, unsupervised and reinforcement learning settings. An extension to this survey identified additional characteristics of these systems, and considered more recent methods (Chen & Liu, 2018). Although interest in lifelong machine learning has been limited, these surveys denote an increasing attention of the machine learning community on systems that learn to learn, as proposed by Thrun (1996).

Continual learning has studied learning systems that learn several tasks continually. The concept was first coined in the context of reinforcement learning. Continual learning was originally defined as the ability of agents to develop complex behaviours without a pre-established end (Ring, 1994). More concretely, continual learning was associated to continual processes where agents learn sequentially, by solving one task at a time (Ring, 1997). The knowledge acquired during this task can be potentially used later to solve a different task. The CHILD system was proposed as an approach to continual learning that satisfied these characteristics (Ring, 1997). More recently, continual learning gained increasing interest in the context of deep neural networks. Research has mainly focused on avoiding *catastrophic forgetting* of knowledge acquired during previous tasks, while learning new tasks and integrating new knowledge into an existing neural network. Parisi et al. (2018) described methods that tackle this problem. The authors identified methods based on: regularization to impose constraints on how the network changes as new tasks are observed, dynamic network architectures that can change as more tasks are observed, *e.g.* by expanding or shrinking subnetworks, and memory management and dual-memories, *e.g.* long-term and short-term memories. The problem of *catastrophic forgetting* was also studied in the context of knowledge consolidation in neural networks (Robins, 1995, 1996), which studied the challenge of integrating new information into a learning system by rehearsing previous tasks. These approaches have been recently extended to deep neural networks (Atkinson et al., 2018a) and deep reinforcement learning (Atkinson et al., 2018b). The possibility of improving knowledge whilst integrating new information was also explored in previous studies on knowledge consolidation (Silver et al., 2015).

Table 1 summarises the approach of each of the paradigms explained above. The main research in each of these areas is also listed. An extended list of existing research can be found in surveys for transfer learning and domain adaptation (Pan & Yang, 2010; Weiss et al., 2016), multitask learning (Caruana, 1998), metalearning (Brazdil et al., 2008; Lemke et al., 2015), lifelong machine learning (Chen & Liu, 2016, 2018) and continual learning (Parisi et al., 2018). Figure 2 provides an intuition into how these areas are connected.

In Sections 5 and 6 we describe specific research that has studied two of the core characteristics of lifelong machine learning systems described by Chen and Liu (2016): 1) learning new tasks using knowledge from previous tasks, 2) learning continuously and incrementally. For the former, we describe methods that transfer directly from previous supervised models or hypothesis to a supervised task. We name this the ability of *transferring forward*. For the latter, we focus on discussing key research proposed under the umbrella of supervised lifelong machine learning and supervised continual learning systems. We describe research that has tackled the problem of refining or retaining existing knowledge, by means of transfer or a similar mechanism. We name this the ability of *transferring backward*.

Table 1: Previous research in Lifelong Machine Learning and related paradigms, for supervised batch learning. Hypothesis Transfer Learning is listed as a paradigm due to the relevance for this paper.

Paradigm	Aim	Approach	Key Research
Transfer Learning and Domain Adaptation	To improve the performance of a target task.	Transfer of data or features from a source domain to a target task.	(Dai, Yang, Xue, & Yu, 2007) (Daumé III, 2009) (Duan et al., 2009) (Glorot, Bordes, & Bengio, 2011) (Gong et al., 2012) (Hoffman et al., 2012) (Bengio, 2012) (Oquab, Bottou, Laptev, & Sivic, 2014) (Yosinski et al., 2014) (Long et al., 2015) (Rusu, Rabinowitz, Desjardins, Soyer, Kirkpatrick, Kavukcuoglu, Pascanu, & Hadsell, 2016) (Li & Hoiem, 2017) (Kirkpatrick, Pascanu, Rabinowitz, Veness, Desjardins, Rusu, Milan, Quan, Ramalho, Grabska-Barwinska, et al., 2017) (Finn et al., 2017) (Ravi & Larochelle, 2017)
Multitask Learning	To improve the performance of a group of tasks.	Learn multiple tasks in parallel, by sharing knowledge.	(Ghosn & Bengio, 1997) (Caruana, 1998) (Baxter et al., 2000) (Kumar & Daume III, 2012) (Ruder, 2017) (Jiang, Wu, Wang, Xue, & Chang, 2018)
Metalearning	To accumulate and use experience over several applications of a learning system.	Accumulate knowledge, and optionally learn, about learning experiences.	(Brazdil et al., 2008) (Lemke et al., 2015) (Santoro et al., 2016) (Finn et al., 2017) (Amit & Meir, 2018) (Yoon et al., 2018) (Rusu et al., 2018)
Hypothesis Transfer Learning	To improve the performance of one or several target tasks.	Transfer of knowledge from a group of available source hypotheses.	(Tommasi et al., 2014) (Kuzborskij et al., 2015) (Mozafari & Jamzad, 2016) (Wang & Hebert, 2016)
Lifelong Machine Learning	To improve the performance of a learning system.	Learn tasks sequentially whilst updating the learning system.	(Thrun, 1996) (Baxter et al., 2000) (Silver & Mercer, 2000) (Thrun & Pratt, 2012) (Silver et al., 2013) (Ruvolo & Eaton, 2013b) (Pentina & Lampert, 2014) (Chen & Liu, 2016) (Chen et al., 2015) (Parisotto, Ba, & Salakhutdinov, 2015) (Fei et al., 2016) (Yoon, Yang, et al., 2017a) (Chen & Liu, 2018)
Continual Learning and Knowledge Consolidation	To retain the performance of a learning system.	Learn tasks sequentially whilst refining a single model.	(Ring, 1994) (Robins, 1995) (Robins, 1996) (Ring, 1997) (Silver et al., 2015) (Atkinson et al., 2018a) (Atkinson et al., 2018b) (Parisi et al., 2018)

3. Preliminaries

In this section we introduce preliminary concepts and notation used in the rest of the paper.

3.1 Support Vector Machines

SVM (Vapnik, 1998) is a technique from the family of statistical learning methods. An SVM function for classification describes an optimal hyperplane that separates classes with the smallest error, whilst maintaining a maximal margin between training examples of different classes. When the training data is not directly separable by a linear function, a kernel function K can be used to map these examples into a feature space of higher dimension where a separating hyperplane exists. This function can be polynomial, radial, sigmoid or some other Mercer kernel, and should be chosen a-priori. The optimal hyperplane can be obtained by solving the problem:

$$\begin{aligned} &\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2 \\ &s.t. \forall i \;\; y_i(\mathbf{w}^\top \phi(x_i) + b) \geq 1 \end{aligned} \tag{1}$$

where the weight vector $\mathbf{w}$ and the bias b are learned parameters. The examples x are transformed into a feature vector using ϕ such that $K(x_i, x_j) = \phi(x_i)^\top \cdot \phi(x_j)$, *i.e.* the kernel function K is obtained from the dot product of these examples transformed using ϕ. The margin (the 'street' separating examples from two classes) of the maximal separating hyperplane is determined by $2/||\mathbf{w}||$. For noisy learning problems, a soft-margin objective is preferred. A soft-margin allows some training examples to violate the margin conditions. The optimal hyperplane for a soft-margin can be obtained by optimizing:

$$\begin{aligned} &\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i \\ &s.t. \begin{cases} \forall i & y_i(\mathbf{w}^\top \phi(x_i) + b) \geq 1 - \xi_i \\ \forall i & \xi_i \geq 0 \end{cases} \end{aligned} \tag{2}$$

where the weight vector $\mathbf{w}$ and the bias b are learned parameters, $\boldsymbol{\xi} = \{\xi_1,...,\xi_n\}$ is a set of slack variables that allow some examples to violate the margin constraints and C is a parameter that controls the compromise between large margin and small margin violations. Figure 3 depicts a toy example of a separating hyperplane for a two-dimensional binary classification task.

Using the method of Lagrange multipliers the formulated primal can be reformulated as the dual (Schölkopf & Smola, 2002):

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n}\alpha_i \;-\; \frac{1}{2}\sum_{i=1,j=1}^{n,n}\alpha_i\alpha_j y_i y_j K(x_i,x_j) \\ &s.t. \sum_{i=1}^{n} y_i\alpha_i = 0, \forall i \; 0 \leq \alpha_i \leq C \end{aligned} \tag{3}$$

The problem is to find a set of coefficients $\alpha_i, 1 \leq i \leq n$, that maximises the objective. Non-zero coefficients correspond to support vectors, training examples defining the boundary

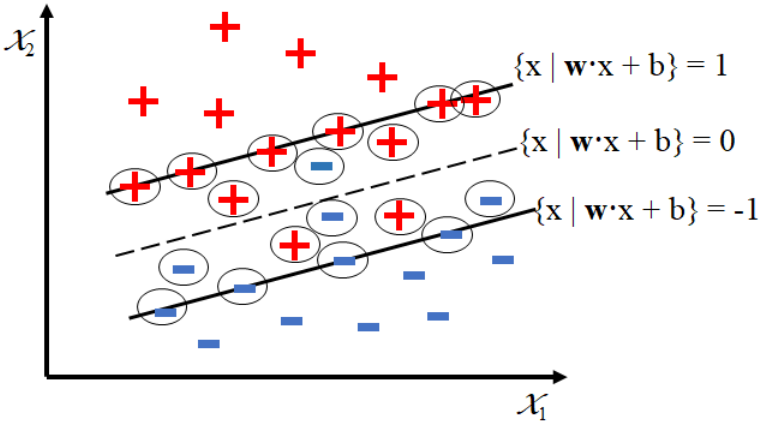


Figure 3: An optimal separating hyperplane for a binary classification problem of two features $\mathcal{X}_1$ and $\mathcal{X}_2$. Circles denote the *support vectors*, training examples defining the decision boundary. Soft-margins allow some support vectors to lie inside the margin, some of which could be misclassified. The separating hyperplane is described by a set of weights or parameters $\mathbf{w}$. An example vector x can be classified using the function $\mathbf{w}x + b$, with b a bias term that is also learned.

that separates the examples from different classes. Support vectors with coefficients strictly within the bounds, *i.e.* $\{\alpha_i | \ 0 < \alpha_i < C\}$, will lie exactly on the margin. Support vectors such that $\{\alpha_i | \ \alpha_i = C\}$ are bounded support vectors and lie inside the margin, some of which could be misclassified. The parameter vector $\mathbf{w}$ for the primal in Eq. 2 can be recovered by:

$$\mathbf{w} = \sum_{i=1}^{n} y_i \alpha_i \phi(x_i) \tag{4}$$

After learned, a SVM hypothesis or decision function is represented by a set of l training examples selected as support vectors, with their corresponding learned coefficients:

$$f^{(c)} = \{(x_1, y_1, \alpha_1), \ldots, (x_l, y_l, \alpha_l)\} \tag{5}$$

which in binary classification problems can be used to classify a new example x, by:

$$f^{(c)}(x) = sgn\left(\sum_{i=1}^{l} \alpha_i y_i K(x_i, x) - b\right) \tag{6}$$

in which case $f^{(c)}(x)$ predicts a value $\{-1, 1\}$. For the purposes of **Proficiente**, different SVM hypotheses learned by the system could use different kernel functions, K.

Existing SVM solvers find a solution to this optimization problem efficiently. Sequential Minimal Optimization, SMO (Platt, 1998), solves the problem heuristically by optimizing (*i.e.* finding the values of coefficients) for two x_i, x_j training examples at a time. These are the training examples with largest violations of the constraints at a given iteration while solving Eq. 3. After x_i and x_j are selected, a gradient projection method is used to decide on their optimal α_i, α_j at the current iteration. This process is repeated until all $\{x_1, \ldots, x_n\}$, and their corresponding $\{\alpha_1, \ldots, \alpha_n\}$, meet the optimality conditions. SVM as a classification technique and SMO as a solver for the SVM problem are a central part of our proposed framework. Bottou and Lin (2007) presented a comprehensive explanation of SVM and SMO.

3.2 Transfer Learning and Hypothesis Transfer Learning

Transfer learning aims to improve learning of a target function $f^{(t)}$ to be learned on a task $T^{(t)}$ by using knowledge from a source domain $\mathcal{D}^{(s)}$ or task $T^{(s)}$. A domain $\mathcal{D}^{(s)}$ consists of both a feature space $\mathcal{X}^{(s)}$ and a marginal probability distribution on the set of random variables $X^{(s)}$ available for learning, $P(X^{(s)})$. A target task consists of a domain $\mathcal{D}^{(t)}$, a sample $Y^{(t)}$ of the label space $\mathcal{Y}^{(t)}$ and the conditional probability $P(Y^{(t)}|X^{(t)})$ which is to be learned from a sample $D^{(t)} = \{(x_1^{(t)}, y_1^{(t)}), \dots, (x_n^{(t)}, y_n^{(t)})\}$, where $\forall i \;\; x_i \in X^{(t)}, y_i \in Y^{(t)}$. In transfer learning, either $\mathcal{D}^{(s)} \neq \mathcal{D}^{(t)}$, because either $\mathcal{X}^{(s)} \neq \mathcal{X}^{(t)}$ or $P(X^{(s)}) \neq P(X^{(t)})$, or $T^{(s)} \neq T^{(t)}$. Different transfer learning scenarios may arise depending on which of these conditions is satisfied (Pan & Yang, 2010).

Hypothesis transfer learning transfers knowledge directly from a set S of source functions or hypotheses $f^{(s)}$ learned on previous tasks to a target task $T^{(t)}$. Apart from this set of source hypotheses, the target task has a set of examples available for learning: $D^{(t)} = \{(x_1^{(t)}, y_1^{(t)}), \dots, (x_n^{(t)}, y_n^{(t)})\}$. Similar to transfer learning, the aim is that the target function $f^{(t)}$ can be learned faster or achieves better performance on unobserved examples. Hypothesis transfer learning is also applicable to scenarios where, for each $f^{(s)} \in S$ learned from a sample $D^{(s)}$ which is no longer available, either $\mathcal{D}^{(s)} \neq \mathcal{D}^{(t)}$, because either $\mathcal{X}^{(s)} \neq \mathcal{X}^{(t)}$ or $P(X^{(s)}) \neq P(X^{(t)})$, or $T^{(s)} \neq T^{(t)}$.

In this paper we explore the hypothesis transfer learning setting where both the source and target tasks are binary classification tasks, *i.e.* $Y^{(s)} \in \{-1, +1\}$ and $Y^{(t)} \in \{-1, +1\}$, the feature space of the source and target tasks is the same, *i.e.* $\mathcal{X}^{(s)} = \mathcal{X}^{(t)}$, but the marginal probability distributions of the source $X^{(s)}$ and the target $X^{(t)}$ are different, *i.e.* $P(X^{(s)}) \neq P(X^{(t)})$.

3.3 The Lifelong Machine Learning Setting

Lifelong machine learning is an area of increasing interest for the machine learning community (Silver et al., 2013; Chen & Liu, 2016, 2018). In lifelong machine learning, a system that learns a sequence of tasks should perform continuous learning by learning new tasks better while storing knowledge increasingly and incrementally (Chen & Liu, 2016). This paper studies the key characteristics of these kinds of systems defined by Chen and Liu (2016): learning new tasks better, performing continuous learning and storing knowledge increasingly and incrementally. We define our lifelong machine learning setting below.

Definition 1. *In our lifelong machine learning setting, a sequence of* $T = \{T_1, \dots, T_N\}$ *supervised tasks is observed by a learning system. These tasks are observed sequentially, one at a time, at given timestamps* $t = \{t_1, \dots, t_N\}$*. Each task is a binary classification task, i.e. the sample class or label space* $Y^{(t)} \in \{-1, 1\}$*, for each task. All tasks are represented in the same feature space, i.e.* $\mathcal{X}_1 = \mathcal{X}_2, \dots, = \mathcal{X}_N$*. The aim of a target task at a specific timestamp* t *is to learn a function or hypothesis* $f^{(t)} : X^{(t)} \to Y^{(t)}$ *using a distribution* $D^{(t)} = \{(x_1^{(t)}, y_1^{(t)}), \dots, (x_n^{(t)}, y_n^{(t)})\}$ *that is sampled i.i.d from an underlying* *probability distribution, and helped by a hypotheses set* $S = \{f_1^{(s)}, \ldots, f_{N-1}^{(s)}\}$ *that has been accumulated in a knowledge base as a result of previous tasks. The performance of each of these hypotheses is measured by a performance metric such as accuracy. Furthermore, existing hypotheses* $f^{(s)} \in S$ *are also expected to benefit from a* $f^{(t)}$ *learned recently.*

This setting is limited to supervised lifelong learning of several binary classification tasks observed sequentially, one at a time. The training examples for these tasks are observed sequentially. These training examples are represented in the same feature space for all tasks. Therefore, in our framework we study the problem of homogeneous transfer (Weiss et al., 2016). The performance of the hypotheses or functions obtained as a result of learning is measured on a set of target examples for each task. Our aim is to study scenarios where the number of tasks is unlimited and does not need to be established a-priori. Tasks received by the system are potentially related to previous tasks, and this relatedness can be determined by means of a standard metric such as for example Kullback-Leibler divergence.

We study long-term learning systems composed of hypotheses stored in the form of SVM hypotheses. A new set of examples received by the system is assumed to correspond to the training examples of a new task that needs to be learned. Therefore, we do not yet solve the problem of identifying when the system is observing a new task, *i.e.* we do not explore the problem of open-world classification described recently by Chen and Liu (2018) and studied in some of the previous research (Fei et al., 2016). We also do not yet explore the problem of training examples from different tasks received at different points in time, such as discussed by Silver and Poirier (2007). The proposed framework performs sequential task learning by learning one class or category at a time, and storing one SVM hypothesis representing the learned task. Each task is a binary classification problem, with examples from the new class as positive examples and examples from other classes as negative examples. Each hypothesis is stored independently, and therefore in this setting there is no need to explore mechanisms for weighting old versus recent knowledge about the same task, such as discussed by Silver, Yang and Li (2013).

4. Overview of a Framework for Knowledgeable Supervised Machine Learning Systems

We propose **Proficiente**, a lifelong machine learning framework that encourages learning of new tasks and refinement of existing knowledge of previous tasks. The framework pursues the following abilities: 1) to learn a non pre-established number of binary classification tasks sequentially, 2) to transfer knowledge to aid learning of new tasks, 3) to refine knowledge of previous tasks, 4) to become more knowledgeable while learning more tasks. These abilities reflect desired characteristics of these kinds of systems envisioned by Chen and Liu (2016), some of which were previously pointed out by Ring (1997) and by Silver, Yang and Li (2013).

Proficiente relies on transferring knowledge across tasks sequentially in two directions: forward to new tasks and backward to existing hypotheses that represent previous tasks. Therefore, **Proficiente** can be categorised as a hypothesis transfer learning approach for tasks learned sequentially. The framework is supported by a knowledge base that stores knowledge to be used in the long-term and knowledge that supports the transfer processes occurring in the system. The proposed framework comprises the instantiation of the scheme

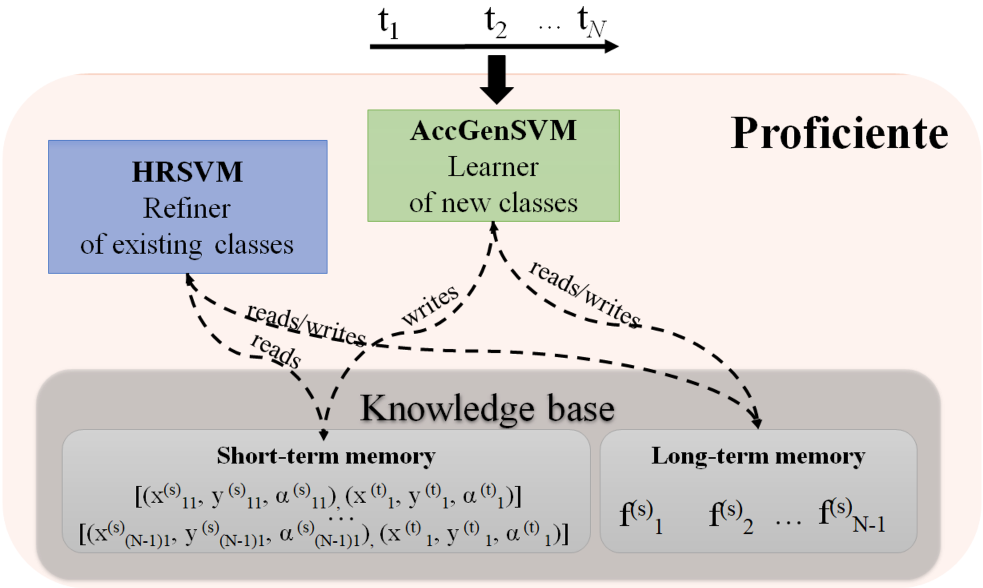

t_1 t_2 … t_N
Proficiente
HRSVM
Refiner
of existing classes
AccGenSVM
Learner
of new classes
reads/writes
reads
writes
reads/writes
Knowledge base
Short-term memory
$[(x^{(s)}_{11}, y^{(s)}_{11}, \alpha^{(s)}_{11}), (x^{(t)}_1, y^{(t)}_1, \alpha^{(t)}_1)]$
…
$[(x^{(s)}_{(N-1)1}, y^{(s)}_{(N-1)1}, \alpha^{(s)}_{(N-1)1}), (x^{(t)}_1, y^{(t)}_1, \alpha^{(t)}_1)]$
Long-term memory
$f^{(s)}_1$ $f^{(s)}_2$ … $f^{(s)}_{N-1}$

Figure 4: A framework for lifelong machine learning using selective transfer of knowledge. Two learners are used for a sequence of tasks $\{t_1, \ldots, t_N\}$: a learner of new classes and a refiner of existing classes. The knowledge base, composed of a short-term and a long-term memory, is read or written as new tasks arrive or existing hypotheses are refined. Short-term memory maintains tuples $[(x^{(s)}, y^{(s)}, \alpha^{(s)}), (x^{(t)}, y^{(t)}, \alpha^{(t)})]$. Long-term memory stores hypotheses $f^{(s)}$ for future use and refinement.

presented in Figure 1. An sketch of **Proficiente** is shown in Figure 4. The framework is composed of:

- A sequence of N supervised learning tasks $T = \{t_1, t_2, \ldots, t_N\}$, observed at different points in time. Each task has a training dataset from which to learn. Each step of the sequence is assumed to correspond to a new task, for which examples are observed at that corresponding time of the sequence.

- **AccGenSVM** (Benavides-Prado et al., 2017), a learner of new binary classification tasks received sequentially. This learner performs selective transfer from previous hypotheses stored in the knowledge base. This learner pursues the first ability of lifelong machine learning systems described by Chen and Liu (2016): learning new tasks better. Details of this learner are provided in Section 5.

- **HRSVM**, a refiner of existing hypotheses that represent previous tasks. This refiner performs selective transfer from a target hypothesis learned recently to existing source hypotheses. This learner pursues a second ability of lifelong machine learning systems described by Chen and Liu (2016): performing continuous learning. Details of this component are described in Section 6.

- A knowledge base that fulfills the third ability of lifelong machine learning systems described by Chen and Liu (2016): store knowledge increasingly and incrementally. This knowledge base stores knowledge of two kinds:
  - Short-term knowledge required for the transfer process. This knowledge is composed of tuples of shared knowledge between a source hypothesis and a target learned recently. **HRSVM** uses these tuples for refinement. This process is described in detail in Section 6. The short-term knowledge is stored in a short-term memory. This memory is written by the learner of new classes during the transfer forward stage. The short-term memory is also read by the refiner of existing hypotheses. After these tuples are collected, the memory can be cleared until the next task is received.
  - Long-term knowledge required for the system to operate. This long-term knowledge is stored in the representational form described by Silver and Poirier (2007), where only the final result of learning is stored in the long-term. This knowledge consists of hypotheses $f^{(s)}$ that are being learned sequentially. Each $f^{(s)}$ is stored independently. Since every hypothesis is learned or refined using SVM, every $f^{(s)}$ is represented as in Eq. 5, *i.e.* as a set $\{(x_1^{(s)}, y_1^{(s)}, \alpha_1^{(s)}), \ldots, (x_l^{(s)}, y_l^{(s)}, \alpha_l^{(s)})\}$, for l support vectors in the corresponding hypothesis. This memory is read by the learner of new classes to collect knowledge while transferring forward. The memory is also written by this same component to store a target hypothesis $f^{(t)}$ learned recently, which will become part of the source hypotheses set during subsequent tasks. During a new task this memory is read by the refiner of existing classes to collect hypotheses $f^{(s)}$ that need to be refined, and written again to save their refined versions. This memory increases linearly with the number of tasks, and therefore the space complexity is $O(Nl)$, for N tasks and l support vectors on each $f^{(s)}$ function learned during these tasks. Each hypothesis is stored in long-term memory as soon as it is learned. Therefore, the efficiency of storage will depend on the particular technology on which this long-term memory is implemented. Similarly for the efficiency of retrieval, which could also be performed in parallel when multiple hypotheses are being retrieved at the same time during learning of a new task. Although **Proficiente** aims to store all the hypotheses learned sequentially, mechanisms for optimising the use of long-term storage could also been implemented (*e.g.* discarding hypotheses that have not been recently retrieved for prediction or for transferring forward to new tasks).

Although **Proficiente** explicitly pursues the abilities of lifelong machine learning systems described by Chen and Liu (2016), the proposed framework also satisfies the essential ingredients of lifelong machine learning systems described previously by Silver, Yang and Li (2013): 1) the retention of learned task knowledge, 2) the selective transfer of prior knowledge when learning new tasks, and 3) a systems approach that ensures the effective and efficient interaction of the retention and transfer elements. In particular, through **HRSVM** the framework aims for effective retention of existing knowledge, by also potentially increasing the accuracy of related prior knowledge. The framework pursues efficient retention by only retaining SVM hypotheses in the long-term memory, while using the short-term memory only during learning of new tasks. Through **AccGenSVM** the framework aims to learn new tasks with at least the same accuracy as when these tasks are learned from their training examples only, by also selecting the most appropriate knowledge to transfer. **AccGenSVM** also aims to reduce the convergence rate while learning these new tasks. The framework retains knowledge in representational form of SVM hypotheses, while performing functional transfer by using support vectors of both previous and new tasks. The framework is aimed to also be scalable to a large and unknown number of tasks.

At the same time, **Proficiente** possesses limitations in terms of some of the challenges of lifelong machine learning systems described by Silver, Yang and Li (2013). The framework is currently limited to the same input types, *i.e.* the same feature representation for all tasks, and to the same output types, *i.e.* binary classification tasks for all the tasks expected by the learning system. The framework does not currently support rehearsal of a task except by the explicit refinement of knowledge of that task using **HRSVM**. The accumulation of practice is given by the sequential learning of multiple binary classification tasks received one after the other, rather than as a continuum that does not distinguish tasks or as a curriculum of tasks that can also be determined by the system.

5. Learning of New Tasks

In this section we describe **AccGenSVM**, a method that transfers selectively from a set of available SVM source hypotheses to a target SVM task (Benavides-Prado et al., 2017). In the proposed method, source support vectors are selected and transferred as privileged information at training time (Vapnik & Izmailov, 2016). Learning with privileged information was proposed as an approximation to knowledge transfer using SVM (Vapnik & Vashist, 2009; Pechyony, Izmailov, Vashist, & Vapnik, 2010; Sharmanska, Quadrianto, & Lampert, 2013; Lapin, Hein, & Schiele, 2014; Niu, Li, & Xu, 2016; Zhou, Xu, Pan, Tsang, Qin, & Goh, 2016; Yan, Nie, Li, Gao, Yang, & Xu, 2016). Similar to classic transfer learning, an algorithm that exploits privileged or additional information aims for faster or more accurate learning on a target task. In previous research, privileged information has been typically represented as additional features that extend the existing target features.

In **AccGenSVM** we proposed to use this information to determine upper-bounds on the coefficients to be learned. Since the privileged information is only required at training time, the proposed method only needs to access the source hypotheses at training time, as opposed to previous hypothesis transfer learning approaches. The relatedness of the source hypotheses with the target training examples is first determined using Kullback-Leibler (KL) divergence. **AccGenSVM** then selects source support vectors related to the target training examples, and transfers coefficients learned for these source support vectors. These coefficients are aggregated and used to upper-bound coefficients to be learned on the target training examples.

In SVM for classification tasks, the coefficients learned for the training examples act as an indication of the importance of these examples for the learned decision boundary (Cristianini & Shawe-Taylor, 2000). **AccGenSVM** selects source hypotheses and transfers learned coefficients as a means to emphasize target training examples that are related to support vectors on these sources. This strategy resembles importance weighting on training data (Lapin et al., 2014), where different training examples are weighted distinctly depending on the relevance for the objective to optimize. The challenge of SVM learning with weighted training data is precisely how to determine these weights. **AccGenSVM** approximates this problem by exploiting related source hypotheses and their support vectors.

This approach is useful when the source data is scarce or difficult to access, and hypotheses related to a target set of examples on a target task are available. Furthermore, the proposed method is especially applicable when the shared knowledge between a source hypothesis and the target examples occurs for only some subspaces of the source hypothesis or target examples. **AccGenSVM** aims to identify these subspaces, which can be potentially used later to determine where shared knowledge occurs. As opposed to previous hypothesis transfer learning approaches, **AccGenSVM** exploits these subspaces of shared knowledge to learn a target hypothesis faster or with a better performance on unobserved examples.

5.1 Previous Research on Transferring Forward from Previous Hypotheses to New Tasks

Hypothesis transfer learning is an alternative for transferring knowledge from previous models or hypotheses learned during previous tasks. Most research in hypothesis transfer learning with SVM has proposed to transfer knowledge from source hypotheses as a whole, without distinguishing fragments of knowledge that can be potentially more useful for a target task (Yang et al., 2007; Aytar & Zisserman, 2011; Tommasi et al., 2014; Kuzborskij et al., 2015; Oneto et al., 2015; Mozafari & Jamzad, 2016; Wang & Hebert, 2016). Furthermore, most existing solutions require the target hypothesis to be represented in terms of both the learned set of parameters and the existing source hypotheses. In this section we describe key solutions to the problem of hypothesis transfer learning using SVM, and the associated challenges.

Adaptive SVM, A-SVM (Yang et al., 2007), is one of the first and most well-known attempts for transferring forward to a target task from a set of existing hypotheses or models. The method uses SVM as the base learner. A-SVM proposed to learn a new SVM hypothesis by regularizing the distance between the set of parameters to be learned on the target task and the set of parameters learned for previous tasks. The SVM objective was adapted to the problem of learning on the target training examples along with an additional term that denotes previous hypotheses. The optimization problem of A-SVM was formulated as:

$$\begin{gathered}\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i \\ s.t. \begin{cases}\forall i \quad y_i \sum_{i=1}^{(T-1)} uf_i^{(s)}(x_i) + y_i(\mathbf{w}^\top\phi(x_i)+b) \geq 1-\xi_i \\ \forall i \quad \xi_i \geq 0\end{cases}\end{gathered} \tag{7}$$

where, similar to Eq. 3, $\mathbf{w}$ is the parameter vector to be learned, $\{\xi_1,\ldots,\xi_n\}$ a set of n slack variables, b is the bias term and C is a trade-off parameter. A-SVM proposed to add the term $y_i\sum_{i=1}^{(T-1)} uf_i^{(s)}(x_i)$ as a constraint that biases learning of $\mathbf{w}$ towards the source hypotheses. The proposed target decision function obtained by A-SVM is represented as:

$$f^{(t)}(x) = \mathbf{w}^\top x + \sum_{i=1}^{(T-1)} uf_i^{(s)}(x) \tag{8}$$

where x is a target example to be classified, $\mathbf{w}$ is the set of parameters learned on the target task using the proposed transfer method, $f_i^{(s)}(x)$ the predictions of the x target example using $(T-1)$ source hypotheses, each denoted as $f^{(s)}$, and $u \in [0,1]$ the user-defined parameter for the importance of each of these sources. Note that the proposed decision function requires to have source hypotheses available at prediction time.

Being the earliest hypothesis transfer learning method, the relevance of A-SVM has been reflected in forthcoming research that tackled some of the challenges faced by this technique. One of these challenges was the inability to distinguish across sources, without relying on prior information such as the proposed user-defined parameter. Further studies such as GreedyTL (Kuzborskij et al., 2015) and MT-SVM (Wang & Hebert, 2016) focused on this problem. The challenge of controlling the trade-off between an increasing SVM margin size and the amount of knowledge transferred was later studied in PMT-SVM (Aytar & Zisserman, 2011). We also explore this challenge in Section 6. A final challenge for A-SVM, which is explored in Section 5.4 of this paper, was the inability to distinguish across fragments of these sources that could potentially be more useful for the target task. Since in hypothesis transfer learning the distributions of the source and target hypotheses do not generally match, identifying and transferring the appropriate subspaces of shared knowledge could potentially make this transfer more precise and helpful for the target task.

GreedyTL (Kuzborskij et al., 2015) studied the problem of finding the best set of source hypotheses for transferring to a target binary classification task. Therefore, GreedyTL aimed to solve one of the challenges posed by A-SVM (Yang et al., 2007). The authors proposed to use a greedy search approach for this purpose. Their solution was based on an L2-regularized version of Forward Regression (Trevor, Robert, & JH, 2009), a feature selection algorithm. Similar to A-SVM (Aytar & Zisserman, 2011), the problem of learning the target hypothesis was formulated as the problem of regularizing the distance between the set of weights or parameters to be learned using target training examples and a linear combination of the parameters of the source hypotheses. The target decision function would have a similar form to A-SVM:

$$f^{(t)}(x) = \mathbf{w}^\top x + \sum_{i=1}^{(T-1)} \beta_i f_i^{(s)}(x) \tag{9}$$

where x is an example to be predicted, $\boldsymbol{w}$ is the set of parameters learned for the target task, and $f^{(s)}$ is a source hypothesis learned on a previous task. There are $(T-1)$ of these source hypotheses. A parameter vector $\boldsymbol{\beta}$ that controls the influence of each source hypothesis is also to be learned. Note that the target hypothesis obtained by GreedyTL is also represented in terms of both the target parameters and the existing hypotheses. Similar to A-SVM, the method assumed that all the subspaces of the source hypotheses were equally relevant for the target task. Therefore, knowledge from these sources was transferred as a whole to the target task.

Projective Model Transfer, PMT-SVM (Aytar & Zisserman, 2011), was proposed as an extension of A-SVM (Yang et al., 2007). Similar to A-SVM, the authors proposed to regularize the difference between the target training examples and a previous hypothesis learned during a related task. Rather than formulating the problem to minimize the difference between the target parameter vector to be learned and the parameter vector from a source hypothesis, PMT-SVM proposed to learn the target weight vector by projecting it onto the source weight vector. The authors also proposed an alternative for transfer that relied on deforming the weights learned for a source hypothesis. PMT-SVM was originally formulated and tested for scenarios where only a single source hypothesis is used for transfer.

Mozafari and Jamzad (2016) studied the problem of hypothesis transfer learning for source and target tasks that use different feature representations. They proposed the Heterogeneous Max-Margin Classifier Adaptation (HMCA) algorithm to solve the problem of transferring from a single source hypothesis on a single dimension rather than in the original hypothesis space. The authors demonstrated that, under specific conditions, the one-dimensional space constructed by applying a source SVM hypothesis or classifier to both the source and target examples represents the common space of the source and the target tasks. The assumption is that the one-dimensional representation of the source and target domains is related. The formulation of the problem was based on A-SVM (Aytar & Zisserman, 2011). The target SVM hypothesis was learned by regularizing the distance between the source and target hypotheses in the proposed one-dimensional space, an idea that is similar to previous hypothesis transfer learning methods that operate in the original hypothesis spaces. The main challenge of the method is the extensibility to scenarios of multiple source hypotheses, since in this case the regularized distance should be formulated to consider multiple source one-dimensional spaces. The proposed metric should also be adapted for considering more than a single source hypothesis.

Wang and Hebert (2016) proposed Model Transfer SVM, MT-SVM, to tackle the problem of selecting source hypotheses from a library of sources. The problem was formulated as a feature selection problem solved using elastic net regularization. Similar to GreedyTL (Kuzborskij et al., 2015), the SVM objective to solve involved regularizing the distance between $\boldsymbol{w}$, the parameter vector learned on the target examples, and a linear combination of the parameters of source hypotheses weighted by a $\boldsymbol{\beta}$ vector which should be also learned. The source selection method was accompanied by a mechanism to generate an expressive set of unsupervised universal source hypotheses that could be used during a target task. The success of the method relies on the generation of the unsupervised universal sources, which requires solving an optimization problem to iteratively find the labels and predictions of the binary codes to be learned (Rastegari, Farhadi, & Forsyth, 2012). The solution to this problem requires training several SVM, one for each iteration of labels and predictions to be optimized for satisfying the objective.

Shrinkage Learning SVM, SL-SVM (Oneto et al., 2015), was proposed as a method aimed to consider a-priori knowledge for training a target SVM hypothesis. The method proposed the use of *hints* to initialize a SVM hypothesis to be learned. The proposed decision function for a target SVM hypothesis $f^{(t)}$ using the proposed method was similar to A-SVM (Yang et al., 2007) and GreedyTL (Kuzborskij et al., 2015):

$$f^{(t)}(x) = g(x) + f^{\delta}(x) \tag{10}$$

where $g(x)$ represents the additional hint that is available during learning of $f^{(t)}(x)$. Note that, similar to A-SVM (Yang et al., 2007) and GreedyTL (Kuzborskij et al., 2015), SL-SVM proposes to use the function $g(x)$ representing these hints at prediction time. Although the use of hints as a mechanism for aiding learning on a target task is a long-standing idea (Abu-Mostafa, 1995), Oneto et al. (2015) explicitly identified the use of a model or hypothesis generated for a related problem as one possibility to generate such hints. Therefore, the problem of shrinkage learning can be categorised as an approximation to hypothesis transfer learning using hints. The problem is formulated as the optimization of the SVM dual which is similar to A-SVM (Yang et al., 2007). The method was originally formulated for scenarios where a single function $g(x)$ can be used to represent hints. Therefore, in the context of hypothesis transfer learning it is limited to transfer from a single source.

Recent research at the intersection of deep learning with transfer learning and deep learning with metalearning has demonstrated that transferring or adapting deep neural networks can improve the performance of one or several target tasks received sequentially. These tasks can also be learned efficiently. This approach looks similar to hypothesis transfer learning research where, rather than transferring from source data, previous models are used to aid new learning tasks. In deep transfer learning generally the objective is to maintain a single network to represent all tasks. Some research, for example, exploit rich initial training data from a variety of classes to learn networks that can be easily transferred to new classes (Bengio, 2012; Oquab et al., 2014; Rusu et al., 2016; Li & Hoiem, 2017). Other methods have proposed to use a rich set of initial tasks to learn highly generalizable parameters that can be quickly adapted in future few-shot learning problems (Finn et al., 2017; Ravi & Larochelle, 2017). As transfer learning methods, these are typically concentrated on improving the performance of target tasks, by transferring forward, especially when the target training data is scarce. The methods are typically focused on transferring from a single source to a target, or to multiple targets observed sequentially.

5.2 Learning with Privileged Information

Learning with privileged information is an option for faster or more accurate learning when additional information is available during training of a new task using SVM (Vapnik & Izmailov, 2016). SVM+ (Pechyony et al., 2010) formulated the problem of learning an SVM with privileged information as the soft-margin primal:

$$\begin{aligned} &\min_{\mathbf{w},b,\tilde{\mathbf{w}},\tilde{b}} \frac{1}{2}\Big(\|\mathbf{w}\|_2^2 + \gamma\|\tilde{\mathbf{w}}\|_2^2\Big) + C\sum_{i=1}^{n}\xi_i(\tilde{\mathbf{w}},\tilde{b}) \\ &s.t. \begin{cases} \forall i \quad y_i(\mathbf{w}^\top\phi(x_i)+b) \geq 1-\xi_i(\tilde{\mathbf{w}},\tilde{b}) \\ \forall i \quad \xi_i(\tilde{\mathbf{w}},\tilde{b}) \geq 0 \end{cases} \end{aligned} \tag{11}$$

where similar to Eq. 2, $\mathbf{w}$ is the parameter vector to be learned and C is a trade-off parameter. The parameter vector $\tilde{\mathbf{w}}$ is learned on the space of the privileged information. The set of slack variables $\boldsymbol{\xi}$ in Eq. 2 is substituted by $\boldsymbol{\xi}(\tilde{\mathbf{w}},\tilde{b})$, which is learned as a function of the privileged information, such that $\boldsymbol{\xi}(\tilde{\mathbf{w}},\tilde{b}) = (\tilde{\mathbf{w}}\cdot\phi(\tilde{x})+\tilde{b})$, with $\tilde{x}$ a training example in the space of privileged features. Finally, γ is a parameter to control the trade-off between the two parameter vectors in the objective. Note that this γ parameter is different to the parameter used for SVM problems with RBF kernels, which is usually given the same parameter name.

The dual function for learning with privileged information is (Pechyony & Vapnik, 2011; Lapin et al., 2014):

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n,n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \frac{1}{2\gamma} \sum_{i=1,j=1}^{n,n} (\alpha_i + \beta_i - C)(\alpha_j + \beta_j - C)\tilde{K}(\tilde{x}_i, \tilde{x}_j)$$
$$s.t. \sum_{i=1}^{n} y_i \alpha_i = 0, \ \sum_{i=1}^{n} (\alpha_i + \beta_i - C) = 0$$
$$\forall i \ \ 0 \leq \alpha_i \leq C + (\alpha_i + \beta_i - C), \ \ 1 \leq i \leq n, \ \ \alpha_i \geq 0, \ \ \beta_i \geq 0 \tag{12}$$

where similar to Eq. 3 the set $\boldsymbol{\alpha}$ corresponds to coefficients to be learned for the training examples. SVM+ proposes to learn an additional set of $\boldsymbol{\beta}$ coefficients on the privileged information space that serve to bias or correct the function learned on the original hypothesis space. The formulation considers the dot product between pairs of examples in the hypothesis space, $K(x_i, x_j)$, and the dot product between pairs of examples in the privileged information space, $\tilde{K}(\tilde{x}_i, \tilde{x}_j)$, obtained by means of selected kernel functions K and $\tilde{K}$, respectively.

5.3 SVM Learning with Weighted Training Data

A connection between learning with privileged information and learning with weighted training data using SVM was demonstrated recently (Lapin et al., 2014). In weighted learning, the influence of a training example (x, y) on the final decision function can be strengthened or softened using weights. The optimization problem for learning a soft-margin SVM hypothesis with weighted training examples in the primal form is:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{n} c_i \xi_i$$
$$s.t. \begin{cases} \forall i \quad y_i(\mathbf{w}^\top \phi(x_i) + b) \geq 1 - \xi_i \\ \forall i \quad \xi_i \geq 0 \end{cases} \tag{13}$$

where similar to Eq. 2 $\mathbf{w}$ is the parameter vector to be learned. A set of slack variables $\boldsymbol{\xi}$ is also learned. A weight vector $\mathbf{c} = \{c_1, \ldots, c_n\}$ determines the weight or importance of each training example. Intuitively, the larger a weight c_i for a x_i training example, the smaller the associated error ξ_i to satisfy the minimisation of the objective. Note that the trade-off parameter of a standard SVM primal problem (C in Eq. 2) is no longer part of the weighted SVM problem. The corresponding dual formulation is:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i \quad - \quad \frac{1}{2} \sum_{i=1,j=1}^{n,n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$
$$s.t. \sum_{i=1}^{n} y_i \alpha_i = 0, \forall i \ 0 \leq \alpha_i \leq c_i \tag{14}$$

where similar to Eq. 3 the set $\boldsymbol{\alpha}$ corresponds to coefficients to be learned for the training examples. Each training example is constrained by a corresponding weight c_i. Lapin et al. (2014) demonstrated that the constraint $\forall i \;\; 0 \leq \alpha_i \leq C + (\alpha_i + \beta_i - C)$ in the dual formulation of SVM+ (Eq. 12) has the effect of controlling the influence of each training example in the final function, similar to the constraint $\forall i \; 0 \leq \alpha_i \leq c_i$ of a weighted SVM (Eq. 14).

5.4 Problem Formulation of Selective Transfer Forward from Previous Hypotheses

Inspired by the ideas of learning using privileged information (Vapnik & Izmailov, 2016) and weighted SVM learning (Lapin et al., 2014), we proposed **AccGenSVM**, an alternative method for selective transfer from source hypotheses learned with SVM (Benavides-Prado et al., 2017). The idea is to transfer selected elements from a set of available source hypotheses. To provide context, we start by a general definition of hypothesis transfer learning.

Definition 2. ***Hypothesis transfer learning*** *Given a set of source hypotheses or functions $S = \{f_1^{(s)}, \dots, f_{N-1}^{(s)}\}$, and a target set of examples $D^{(t)} = \{(x_1^{(t)}, y_1^{(t)}), \dots, (x_n^{(t)}, y_n^{(t)})\}$, hypothesis transfer learning aims to use one or more $f^{(s)} \in S$ to aid learning of $f^{(t)}$ on a target task $T^{(t)}$. This task is trained using both $f^{(s)}$ and $D^{(t)}$.*

Definition 2 is general and applies to our proposed method and to existing research in hypothesis transfer learning. We now introduce a formal definition of selective hypothesis transfer between tasks learned using SVM.

Definition 3. ***Selective hypothesis transfer with SVM*** *Given a source hypotheses set $S = \{f_1^{(s)}, \dots, f_{N-1}^{(s)}\}$, where each hypothesis $f^{(s)}$ is represented as in Eq. 5, such that $f^{(s)} = \{(x_1^{(s)}, y_1^{(s)}, \alpha_1^{(s)}), \dots, (x_l^{(s)}, y_l^{(s)}, \alpha_l^{(s)})\}$, and target examples $D^{(t)} = \{(x_1^{(t)}, y_1^{(t)}), \dots, (x_n^{(t)}, y_n^{(t)})\}$, selective hypothesis transfer learning with SVM aims to transfer selected knowledge of the form $(x_i^{(s)}, y_i^{(s)}, \alpha_i^{(s)}) \in f^{(s)}$, $1 \leq i \leq l$, from a subset F of source hypotheses $f^{(s)} \in S$. The aim is to learn a set $\boldsymbol{\alpha} = \{\alpha_1^{(t)}, \dots, \alpha_n^{(t)}\}$ on the target task $T^{(t)}$, faster or with higher accuracy on unobserved samples.*

AccGenSVM identifies a subset $F \subseteq S$ of SVM source hypotheses from which to transfer selected knowledge to the target SVM task. This subset is selected based on the relatedness of each source hypothesis $f^{(s)} \in S$ to the target training examples. This relatedness is measured using KL-divergence. Once F is identified, related source support vectors $x_i^{(s)}$, $1 \leq i \leq l$, for a particular $f^{(s)}$ are also identified for each training example $x_i^{(t)}$, $1 \leq i \leq n$, on the target task. Fragments of source hypotheses $f^{(s)} \in F$ are then transferred by extracting $\alpha_i^{(s)}$, $1 \leq i \leq l$, coefficients which are used to upper-bound coefficients to be learned on the target task, for the corresponding $x^{(t)}$. As a result, training examples which are more resembled by selected source support vectors $x_i^{(s)}$ are given more importance while learning $f^{(t)}$, and contribute more to the objective to optimize. The learning problem in the primal representation is similar to Lapin et al. (2014):

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C \sum_{i=1}^{n} c_i \xi_i$$
$$s.t. \begin{cases} \forall i \quad y_i(\mathbf{w}^\top \phi(x_i) + b) \geq 1 - \xi_i \\ \forall i \quad \xi_i \geq 0 \end{cases} \tag{15}$$

where a set of weights $\mathbf{c} = \{c_1, \ldots, c_n\}$ for the training examples are to be considered as an indication of the importance of these examples for the problem at hand. **AccGenSVM** considers the trade-off parameter C, which serves as the default upper-bound on target training examples which are not subject to transfer from sources. The corresponding dual formulation is:

$$\begin{gathered} \max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i \;-\; \frac{1}{2} \sum_{i,j=1}^{n,n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ s.t. \sum_{i=1}^{n} y_i \alpha_i = 0, \forall i \; 0 \leq \alpha_i \leq C + c_i, c_i = \frac{|F|}{|S|} \sum_{k=1}^{s_i} \alpha^{(s)}{}_{ik} \end{gathered} \tag{16}$$

which uses training examples $(x_i, y_i), 0 \leq i \leq n$, to learn a weight vector $\mathbf{w}$ of the function separating these examples. A set of slack variables $\boldsymbol{\xi} = \{\xi_1,...,\xi_n\}$ allows some examples to violate the margin constraints. As a modification to the original SVM objective (Bottou & Lin, 2007), the upper-bound of a coefficient α_i is constrained by $C + c_i$. This constraint is composed of the original upper-bound C and c_i, an aggregation of $\alpha^{(\mathbf{s})} = \{\alpha_1^{(s)}, \ldots, \alpha_s^{(s)}\}$ coefficients transferred from s_i source support vectors related to a target $x_i^{(t)}$ under evaluation. A factor that accounts for the number of $f^{(s)}$ contributing to the target task, $|F|/|S|$, is also considered.

Selecting which $\alpha_{ik}^{(s)}$ coefficients are transferred requires two steps: 1) identifying the subset F of source hypotheses $f^{(s)} \in S$ from which to transfer, and 2) identifying $x^{(s)}$ source support vectors from each selected $f^{(s)}$ for transfer to x_i. For the first step we proposed to calculate the KL-divergence from the distribution of the source hypothesis $f^{(s)}$, which we name $D^{(f^{(s)})}$, to the distribution of the target training data $D^{(t)}$, between corresponding classes in binary classification problems. KL-divergence is one of the mechanisms to determine relatedness of a pair of distributions before transfer (Pan, Kwok, & Yang, 2008; Zhuang, Cheng, Luo, Pan, & He, 2015). The KL-divergence in the proposed setting is:

$$KL(D^{(t)}||D^{(f^{(s)})}) = D^{(t)} \log \left(\frac{D^{(t)}}{D^{(f^{(s)})}} \right) \tag{17}$$

If $KL(D^{(t)}||D^{(f^{(s)})}) \leq \tau$, with τ a threshold, the probability distributions underlying $D^{(f^{(s)})}$ and $D^{(t)}$ can be assumed to be related, since the loss in information when using $D^{(f^{(s)})}$ to approximate $D^{(t)}$ is small, based on the threshold τ. This threshold encourages transfer to the target task. In practice, this threshold can be selected using techniques such as grid-search over a range of values. A second step selects source support vectors $x^{(s)} \in f^{(s)}$ that are related to a target example $x^{(t)} \in D^{(t)}$ which is being evaluated as part of the solution to the problem in Eq. 16, using k-nearest neighbours. SMO (Platt, 1998) is used as the solver of this problem. A pair of target training examples are evaluated at a given iteration. The upper-bound for each of these training examples is calculated using the proposed method.

5.5 Theoretical Properties of Selective Transfer Forward from Previous Hypotheses

Kuzborskij et al. (2017) demonstrated the effects of transferring from a large set of source hypotheses to a target task. The authors focused on analysing the generalisation properties of hypothesis transfer learning problems that learn functions of the form in Eq. 9:

$$f^{(t)}(x) = \mathbf{w}^\top x + \sum_{i=1}^{(T-1)} \beta_i f_i^{(s)}(x) \tag{18}$$

where a set of source hypotheses $f_i^{(s)}, 1 \leq i \leq (T-1)$, are used along with a target set of parameters $\mathbf{w}$ for predicting a target example x. The relevance of these sources for the target task is determined by a parameter vector $\boldsymbol{\beta}$, which is also learned. The authors demonstrated that, given a set of source hypotheses that are beneficial for a target task, generalisation can occur at the fast rate of $\mathcal{O}(1/n)$, with n the number of examples on the target task, as opposed to the usual $\mathcal{O}(1/\sqrt{n})$ in the setting of empirical risk minimisation. These bounds, however, only consider hypothesis transfer learning methods that learn a parameter vector $\boldsymbol{\beta}$ as an indication of the relevance of the sources for the target task.

Lapin et al. (2014) analysed the equivalence of the solutions achieved by solving the problems of training an SVM hypothesis with weighted training examples and SVM+ (Pechyony et al., 2010) as the SVM implementation of the idea of learning using privileged information (Vapnik & Vashist, 2009). The authors demonstrated that any solution obtained by SVM+ can be also obtained via SVM with weighted training examples, given appropriate weights. Similar to the bounds demonstrated by Kuzborskij et al. (2017) for hypothesis transfer learning of the form in Eq. 18, Pechyony and Vapnik (2010) demonstrated that when the privileged information is beneficial to a target task, a solution for the target task can be found at the rate of $\mathcal{O}(1/n)$ as opposed to the usual rate of $\mathcal{O}(1/\sqrt{n})$. This rate can be achieved even for decision functions that are hard to learn, given a not too small set of training examples. Such a rate is derived from the fact that the privileged information acts as correcting information over the training examples, such that the loss on the original space is bounded by the privileged information as:

$$\mathcal{L}(\hbar(x), y) \leq \mathcal{L}(\varphi(\tilde{x}), y) + C[\mathcal{L}(\hbar(x), y) - \mathcal{L}(\varphi(\tilde{x}), y)]_+ \tag{19}$$

where $\hbar \in \mathbb{H}$ is a hypothesis in the hypotheses space $\mathbb{H}$ and $\mathcal{L}(\varphi(\tilde{x}), y)$ is the loss in the space of the privileged information. Therefore the expected risk of a chosen $\hbar^\Diamond$ is bounded by the empirical risk:

$$R(\hbar^\Diamond) \leq E\{\mathcal{L}(\hbar^\Diamond, y)\} \leq C \cdot E\{\mathcal{L}'((\hbar^\Diamond, \varphi^\Diamond), (x, \tilde{x}, y))\} = C \cdot R'(\hbar^\Diamond, \varphi^\Diamond) \tag{20}$$

where $\mathcal{L}'((\hbar^\Diamond, \varphi^\Diamond), (x, \tilde{x}, y)) = 1/C \cdot \mathcal{L}(\varphi(\tilde{x}), y) + [\mathcal{L}(\hbar(x), y) - \mathcal{L}(\varphi(\tilde{x}), y)]$ is the loss of the selected hypothesis $(\hbar^\Diamond, \varphi^\Diamond)$ that considers both the original space and the space of the privileged information for an example x, *i.e.* $(x, \tilde{x}, y)$.

5.6 Algorithm for Selective Transfer Forward

The algorithmic solution to Eq. 16 is composed of two phases. The first phase is intended to select source hypotheses and source support vectors for transfer, from a possibly large number of these sources. The second phase transfers source coefficients and obtains new upper-bounds for the target training examples. The proposed approach is detailed in Algorithm 1, and works as follows.

Phase 1 - Selection of source support vectors for transfer.

1. Based on SMO as the solver (Bottou & Lin, 2007), **AccGenSVM** selects a pair of candidate target training examples, $x_i^{(t)}$ and $x_j^{(t)}$. At every iteration, the proposed method will gather privileged information of the form specified in Eq. 16 from existing source hypotheses for these two training examples.

2. The set S of source hypotheses is filtered by measuring relatedness of each of the source hypotheses $f^{(s)} \in S$ with the target training data by means of KL-divergence, according to Eq. 17. Every source hypothesis $f^{(s)} \in S$ is compared to the target distribution $D^{(t)}$. The subset of source hypotheses below the specified threshold τ is selected as the subset F of source hypotheses to consider in further steps. The parameter τ can be set based on standard parameter selection methods such as grid-search.

3. A fast k-nearest neighbour method, FNN (Beygelzimer, Kakadet, Langford, Arya, Mount, & Li, 2013), is used to find the source support vectors which are more closely related to the target training examples determined in Step 1, using the subset of source hypotheses selected in Step 2. For each of these sources, a number of source support vectors is selected using FNN. The coefficients of these source support vectors, $\alpha^{(s)}$, are aggregated and will contribute to the upper-bounds for the corresponding target training examples.

Phase 2 - Transfer.

1. Based on the output of Phase 1, **AccGenSVM** calculates the upper-bounds $c_i^{(t)}$ and $c_j^{(t)}$, for the corresponding $\alpha_i^{(t)}$ and $\alpha_j^{(t)}$ to be learned for the target training examples $x_i^{(t)}$ and $x_j^{(t)}$, respectively, as indicated in the modified constraint of Eq. 16.

2. **AccGenSVM** will also consider the number of source hypotheses related to the target task, *i.e.* the size of $F \subseteq S$ of related source hypotheses, to balance the modified upper-bounds. The more source hypotheses contributing to an upper-bound, the higher this upper-bound will be for the corresponding target training example. The regularization parameter C, which can be set using standard parameter selection methods such as grid-search, is also considered.

The computational complexity of training an SVM using SMO can be of $i \times O(n)$, with i iterations and n target training examples, when data is cached (Chang & Lin, 2011). The FNN step to find related source support vectors for a target training example increases this to a worst case $i \times O(n \times \log(N-1))$, for $(N-1)$ source hypotheses. Here n depends on the number of points evaluated, which is smaller for faster convergence rates. Selecting related source hypotheses adds a constant $O(S)$ that depends on the number of available source hypotheses. In terms of convexity, the proposed algorithm behaves like training an SVM with weighted training data (Lapin et al., 2014).

Algorithm 1: Pseudo-code for transfer with **AccGenSVM**

```
Data: Target training examples $D^{(t)} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$; source hypotheses
      set $S$; $F \leftarrow \emptyset$; $x_i, x_j \leftarrow$ selected target training examples by working set
      selection (SMO); $C \leftarrow$ regularization parameter; $\tau \leftarrow$ threshold for
      KL-divergence; $y_i, y_j \leftarrow$ corresponding classes for selected target examples;
      $c_i, c_j \leftarrow 0$; $\alpha_i^{up}, \alpha_j^{up} \leftarrow C$
Result: Upper-bounds of coefficients to be learned $\alpha_i^{up}$, $\alpha_j^{up}$
// Phase 1
forall $f^{(s)}$ in $S$ do
    if $KL(D^{(t)}||D^{(f^{(s)})}) \leq \tau$ (Eq. 17) then
        forall $x^{(s)}$ in $f^{(s)}$ do
            if $x^{(s)}$ and $x_i$ are k-nearest neighbours and $y^{(s)} = y_i$ then
                $c_i \leftarrow c_i + \alpha^{(s)}$
            end
            if $x^{(s)}$ and $x_j$ are k-nearest neighbours and $y^{(s)} = y_j$ then
                $c_j \leftarrow c_j + \alpha^{(s)}$
            end
        end
    end
    $F \leftarrow \{F \cup f^{(s)}\}$
end
// Phase 2
$\alpha_i^{up} \leftarrow C + \frac{|F|}{|S|} c_i$
$\alpha_j^{up} \leftarrow C + \frac{|F|}{|S|} c_j$
```

5.7 Discussion

In this section we explained **AccGenSVM**, a method to transfer forward from previous SVM hypotheses to a target SVM task (Benavides-Prado et al., 2017). The method differs from existing hypothesis transfer learning SVM methods in terms of what, when and how to transfer. **AccGenSVM** transfers selected coefficients from previous SVM hypothesis, when these hypotheses are related with the target task according to the KL-divergence metric. Coefficients learned for source support vectors from related source hypotheses are then transferred as upper-bounds on coefficients to be learned on the target task. Although this selective method achieved similar accuracy as competing hypothesis transfer learning methods, the convergence rate on the target task can be fastened due to higher upper-bounds for target training examples that are relevant for that task. Table 2 summarises the research methods described at the beginning of this section and **AccGenSVM**. We explain aspects of what, when and how these methods perform hypothesis transfer learning.

Table 2: Hypothesis transfer learning in previous research and **AccGenSVM**.

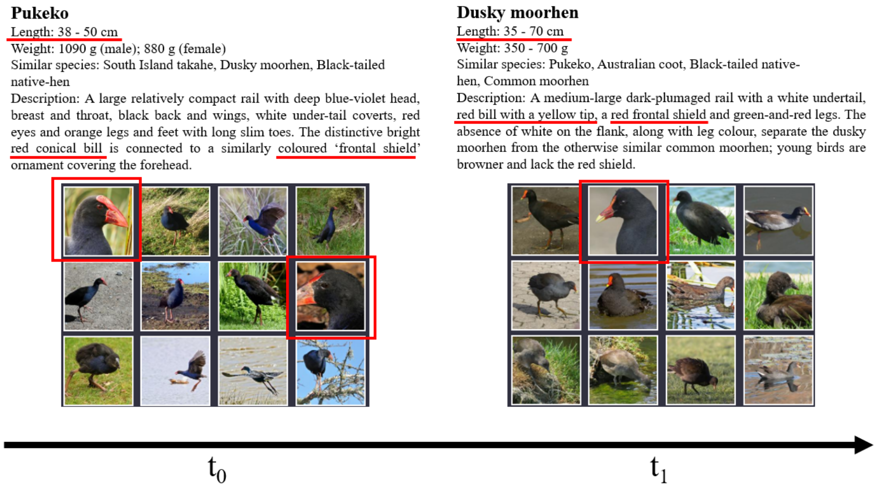
Method	What to transfer	When to transfer	How to transfer
A-SVM (Yang et al., 2007)	Source coefficient vectors as a whole.	When a source hypothesis is a good predictor of target data.	As an additional term that represents a linear combination of previous hypotheses.
PMT-SVM (Aytar & Zisserman, 2011), similarly in (Oneto et al., 2015)			Directly, one source to one target.
MMKT (Tommasi et al., 2014),		As indicated by an additional term that minimizes the leave-one-out error.	As an additional term along with a linear combination of previous hypotheses.
GreedyTL (Kuzborskij et al., 2015), used by (Valerio et al., 2016)		As indicated by a greedy search on the hypotheses set.	As new features, and an additional term for their total contribution.
MT-SVM (Wang & Hebert, 2016)		As far as required by the optimization procedure.	As an additional term that needs to be learned.
HMCA (Mozafari & Jamzad, 2016)	One-dimensional vector.	As indicated by a similarity measure between source and target data.	Directly in one dimension.
Learning with hints (Oneto et al., 2015)	Hints.	Not explicit.	By including the hints as a component of the function to be learned.
Deep transfer learning *e.g.* (Yosinski et al., 2014) (Long et al., 2015)	Network weights.	Not explicit.	By initializing the network with previous weights.
AccGenSVM (Benavides-Prado et al., 2017)	Selected source SVM coefficients.	When a source SVM hypothesis is related to the target training examples.	By upper-bounding target training coefficients using source SVM coefficients.

6. Refinement of Existing Knowledge on Previous Tasks

A natural next step after transferring forward from a set of source hypotheses to a target task is to aim to improve knowledge of the existing hypotheses. This is fundamental for systems to become more knowledgeable while observing more related tasks (Thrun, 1996; Chen & Liu, 2016). Previous lifelong learning research proposed to do this implicitly by maintaining a layer of shared knowledge between tasks, that is refined after each new task (Ruvolo & Eaton, 2013b). Other research proposed to learn continuously by explicitly retraining hypotheses describing previous tasks after learning a target task (Fei et al., 2016). Research in continual learning has proposed to retrain a deep network sequentially as new tasks are observed, whilst avoiding *catastrophic forgetting* of knowledge about old tasks (Parisi et al., 2018).

In this section we formulate the problem of refining an existing hypothesis or function by means of transfer. We envision a second phase of learning focused on refining existing hypotheses by transferring from a target hypothesis learned recently. We name this stage of the lifelong learning process **transfer backward**, since it operates in the opposite direction to regular transfer. The method described in this section, combined with the method presented in Section 5, are applicable to learning systems that observe binary classification tasks sequentially whilst aiming to improve performance of the learning system as a whole. These tasks are learned with SVM.

A simple example application of these kinds of systems is provided in Figure 5. In this example, a supervised learning system has to learn to classify data examples of different and possibly related species of birds. The figure exemplifies two of these classes. Training examples of different species may be observed at different points in time, or across different locations. Each learning task is aimed to learn to classify examples into one of these species. The data that is collected for categorising birds includes: physical characteristics of these birds such as their length, width, color, plumage, tail shape, and several other descriptors, together with the sounds that these animals emit and images of these birds. Some of the examples in these species may have similar characteristics since they belong to the same family of birds. The example in Figure 5 remarks two species: *Pukeko* and *Dusky Moorhen*, of the *Rallidae* family. Suppose that the first task is to learn to classify birds into the *Pukeko*



Pukeko
Length: 38 - 50 cm
Weight: 1090 g (male); 880 g (female)
Similar species: South Island takahe, Dusky moorhen, Black-tailed native-hen
Description: A large relatively compact rail with deep blue-violet head, breast and throat, black back and wings, white under-tail coverts, red eyes and orange legs and feet with long slim toes. The distinctive bright red conical bill is connected to a similarly coloured 'frontal shield' ornament covering the forehead.

Dusky moorhen
Length: 35 - 70 cm
Weight: 350 - 700 g
Similar species: Pukeko, Australian coot, Black-tailed native-hen, Common moorhen
Description: A medium-large dark-plumaged rail with a white undertail, red bill with a yellow tip, a red frontal shield and green-and-red legs. The absence of white on the flank, along with leg colour, separate the dusky moorhen from the otherwise similar common moorhen; young birds are browner and lack the red shield.

Figure 5: An example of a sequential learning system of two classes. This information has been extracted from The Digital Encyclopaedia of New Zealand Birds (Museum of New Zealand Te Papa Tongarewa (Te Papa) & of Conservation, 2019a, 2019b). Similar characteristics for these species of birds are highlighted in red.

category (t_0). The only information available for learning a hypothesis that categorises birds into this species are training examples of birds for which it is known if these are *Pukeko* or not. However, this is not the case for the second task (t_1). Learning to classify *Dusky Moorhen* can be aided by knowledge extracted while learning about *Pukeko*, since these two categories share characteristics as highlighted in Figure 5. For example, one characteristic shared by *Pukeko* and *Dusky Moorhen* is their red bill. *Pukeko* have a conical and fully red bill, whilst *Dusky Moorhen* have a yellow tip. Therefore, refining knowledge about the *Pukeko* category after learning about the *Dusky Moorhen* category can potentially help to reduce error while classifying examples of this species. As a result of refinement, the hypothesis previously learned for *Pukeko* will not classify as *Pukeko* those birds with a red bill and a yellow tip.

We have previously proposed **L3SVM** (Benavides-Prado et al., 2019), a method that aims to refine knowledge of existing hypotheses by transferring backward from a target hypothesis learned more recently. Similar to existing lifelong learning methods, **L3SVM** included a mechanism to control the rate of transfer in order to prevent negative transfer. This mechanism, which is represented as a parameter that decreases while more tasks are learned, depends on the number of tasks that the system expects to observe. However, in long-term learning systems the number of tasks may not be pre-established. Moreover, in these kinds of systems the process of refining existing knowledge should also consider retention of knowledge in the long-term, regardless of the number of tasks. Refinement and retention are desired properties of long-term learning systems (Silver & Poirier, 2007; Silver et al., 2013), although these are usually competing objectives. This is a trade-off which is also observed in biological systems (Bremner et al., 2012). Recent research in continual learning has studied the problem of avoiding *catastrophic forgetting* of old tasks (Parisi et al., 2018; Atkinson et al., 2018a, 2018b). This challenge is similar in nature to the retention challenge recognised for lifelong learning systems (Silver & Mercer, 2000; Silver et al., 2013). Continual learning has revived the interest of studying the problem of *catastrophic forgetting* which was previously studied in the context of knowledge consolidation in neural networks (Robins, 1995, 1996). Some of this work also pointed to the possibility of improving knowledge of previous tasks as a result of learning new tasks (Silver et al., 2015). Nevertheless, the problem of refining existing hypotheses whilst controlling for retention of knowledge, although fundamental, has received very limited attention.

In this section we describe **HRSVM**, a method for transferring selected knowledge backward from a target hypothesis learned recently to related source hypotheses. The aim is to refine existing hypotheses using knowledge collected during a target task which is executed after these sources were learned, whilst aiming for retention of the existing knowledge on these hypotheses. Similar to **L3SVM**, the proposed approach uses sets of source and target support vectors identified while learning the target task as training examples for learning representations of subspaces of shared knowledge between a source and the target. We use **AccGenSVM** to identify these support vectors. This strategy emphasizes refinement towards the knowledge shared between the source and the recent target, particularly when knowledge sharing occurs across different subspaces of these hypotheses. **HRSVM** solves a modified ν-SVM problem by considering the original source hypothesis space along with intermediate representations learned on these subspaces, whilst controlling for retention of existing knowledge. By means of ν-SVM we propose to control the amount of knowledge transfer whilst controlling the margin size. This is the practical reflection of the refinement vs. retention trade-off when transferring across SVM hypotheses or models (Aytar & Zisserman, 2011). To the best of our knowledge, we are the first to propose selective hypothesis refinement with retention.

6.1 Previous Research on Transferring Backward

Explanation-Based Neural Networks, EBNN (Thrun, 1996), is the earliest supervised learning study officially categorised as a lifelong machine learning method. EBNN tackles the problem of learning a *theory of the domain*, a piece of meta-knowledge that summarises properties of a learning system. This meta-knowledge is composed of several functions or hypotheses obtained during learning experiences of the system. These functions are composed of training examples that were used to learn each corresponding function. Therefore, knowledge at the meta-level is composed of several examples coming from several learned functions. These examples are named the *support sets* of these functions.

Given a set of hypotheses stored at the meta-level, EBNN proposes to exploit this knowledge to aid learning of a new task. EBNN identifies and learns invariances shared by the support sets, and uses these invariances to delimit the space of hypotheses that needs to be explored on the target task. The knowledge extracted from these invariances is stored as a neural network. These explanations are used to aid learning of the new task. This is achieved by deriving slope constraints of the target training examples with respect to the invariance network. The main challenge posed by EBNN was the requirement to maintain the proposed invariance network perpetually. Since new tasks may bring new knowledge about the theory of the domain described by the invariance network, the network would also need to be re-trained after learning each new task. The method was mainly focused on improving the performance of a target task, rather than on improving the performance of the system as a whole or on refining knowledge of existing hypotheses or functions. Nevertheless, the relevance of the method relies on being the pioneer study for the problem of learning tasks sequentially one after the other.

An Efficient Lifelong Learning Algorithm, ELLA (Ruvolo & Eaton, 2013b), was proposed as an alternative that extends GO-MTL (Kumar & Daume III, 2012), a multitask learning approach, to the problem of learning tasks sequentially. ELLA relies on learning, updating and maintaining a set of parameters shared by all classes. A new function for a target task was proposed to be learned as a combination of this shared set and a set of parameters that are specific for that task. The problem was formulated to encourage sparseness of the shared set of parameters. Learning a new classification task involved three steps: 1) to learn the target hypothesis, by solving an optimization problem that includes the current shared set of parameters; 2) to relearn this shared set, while optimizing the performance (decreasing the error) of all the tasks learned so far; 3) to recalculate parameters or weights of each task with respect to the shared set of parameters. Since the shared set is updated whenever a new task is observed, the refinement of this set is expected to refine knowledge of classes already learned by the system. Therefore, ELLA tackles the problem of transferring backward from a hypothesis learned recently implicitly. Similar to existing hypothesis transfer learning methods, ELLA does not require access to data from source hypotheses to perform transfer. However, the refinement of the shared set of parameters and the recalculation of the weights of each hypothesis with respect to this set need to be performed after learning each new task. ELLA also poses a challenge similar to EBNN: the shared set of parameters composes knowledge that needs to be maintained perpetually. For classification tasks, the method was originally implemented using a logistic regression as the base learner. Nevertheless, after EBNN this method remains as a reference technique in lifelong machine learning, for both regression and classification tasks.

Cumulative Learning (Fei et al., 2016) is a method that identifies new classes to be learned, learns functions describing these classes and updates existing knowledge. The approach implements the concept of *open-world classification* (Fei & Liu, 2016), a paradigm that relies on maintaining an *unknown* category along with existing categories or classes. The unknown class is fed at test time, whenever the system receives a new example that cannot be classified in any of the existing categories. When the unknown class has accumulated enough examples, two steps need to be performed: 1) to learn a function describing the new class, using examples from that class as the positive examples and examples from other classes as the negative examples, 2) to relearn existing functions or hypotheses, by adding examples from the new class as part of the current negative class and retraining these hypotheses from scratch. Therefore, the method implements the concept of transferring backward explicitly. A similarity metric between existing hypotheses and the hypothesis learned recently is used to decide which source hypotheses to update. Similar to existing hypothesis transfer learning methods for transferring forward, the similarity metric used to decide if an existing hypothesis should be retrained depends on how well this hypothesis predicts target training data. The main challenge faced by CL is the need to store data of all previous tasks for future retraining of the corresponding hypotheses. Furthermore, since the training data for retraining each previous task includes the examples of every new task, the cost of relearning source hypotheses increases as more tasks are learned. Nevertheless, the relevance of this method relies on the exploration of the problem of *open-world classification* in the context of tasks learned sequentially. The ability to identify when new classes should be learned has been recently considered as a desired characteristic of lifelong machine learning systems (Chen & Liu, 2018).

Continual learning has revived as an approximation to lifelong learning using deep neural networks. Parisi et al. (2018) investigated the challenges of continual learning, remarking the need to avoid *catastrophic forgetting* or *catastrophic interference* of previous knowledge. This phenomenon refers to situations when new knowledge collected during recent tasks eliminates or corrupts existing knowledge acquired during previous tasks. This is a long-lasting concern that was also investigated in the context of neural and connectionist networks (McCloskey & Cohen, 1989), and in the context of consolidation of knowledge while learning new tasks (Robins, 1995, 1996).

Recently, Progressive Neural Networks, PNN (Rusu et al., 2016), proposed to use lateral connections of features learned for previous tasks to aid learning of a new task, while preventing catastrophic forgetting of existing knowledge. This method was originally proposed for reinforcement learning problems. The authors proposed to augment an existing network whenever a new task was received. A network for the new task was learned using features learned for the previous tasks, by laterally connecting these features to the features to be learned on the target task. As new tasks arrive, new layers are appended to the existing network. Transfer of knowledge between previous tasks and the target task occurs by the proposed lateral connections. However, these lateral connections are designed in a single *forward* direction, and therefore no changes or refinement would occur in the layers representing previous tasks.

Kirkpatrick et al. (2017) proposed Elastic Weight Consolidation (EWC) as an approximation to the problem of catastrophic forgetting of old tasks. The proposed algorithm is based on slowing down learning of selected network weights based on how relevant these weights are for previous tasks. During the training of the new task, the objective is to optimize for this new task while retaining relevant network weights unchanged. Similar to PNN, EWC is focused on maintaining the performance of previous tasks by freezing the relevant weights. Therefore, previous tasks would not be improved given new knowledge collected in more recent tasks.

Dynamically Expandable Neural Networks, DEN (Yoon et al., 2017a), is a recent method that studies the problem of catastrophic forgetting of old tasks using deep neural networks. The method is conceived in three phases: 1) selective retraining, 2) dynamic network expansion, 3) split and duplication of the existing network. During selective retraining, if a network already exists, a sparse linear model is fitted to predict the incoming task examples using the topmost hidden units of the current neural network. As a result, a subnetwork with units and weights connected to the new task can be identified. This subnetwork is retrained using an element-wise L2 regularizer. During the first task learned by the system, *i.e.* when no network exists yet, an initial network is trained with L1-regularization to promote sparsity. In the second phase, the network is dynamically expanded to accommodate features that are applicable to the new task and which can not be represented by retraining the existing network only. The authors proposed to use group sparse regularization to add a given number of units to each layer. This phase is only performed if the loss of the new task is below a pre-established threshold, a fact that indicates that the retrained network is not sufficient to represent the new task. During the last phase, units are duplicated whenever these seem to be suboptimal for a particular task. To determine this, the amount of catastrophic forgetting of each unit for each task is measured using the L2-distance between a previous task and the new incoming task. If this amount is above a threshold, the corresponding unit is copied twice or more times, depending on the number of previous tasks affected by the change in that unit. After this duplication occurs, the network needs to be retrained fully again, in order to learn new values for these units that are optimal for the previous tasks. It was demonstrated experimentally that DEN is more stable at retaining AUC of previous tasks than other methods. However, similar to PNN and EWC, the method does not improve the performance of the neural network as more tasks are learned. An additional challenge posed by DEN is the requirement to maintain data from previous tasks available for the duplication phase. This poses a drawback for systems composed of hundreds or thousands of tasks, such as long-term learning systems, since it may lead to scalability problems for neural network architectures that become very large, as discussed by Parisi et al. (2018). We provide an experimental example of this in Section 6. Additional recent research has proposed a similar approach to DEN for adapting or expanding the structure of an existing network as more tasks are learned (Li, Zhou, Wu, Socher, & Xiong, 2019). More recently, OWM (Zeng, Chen, Cui, & Yu, 2019) was proposed as an alternative for continual learning that learns new tasks whilst avoiding *catastrophic forgetting* of previous knowledge by restricting new weights to a set which is orthogonal to existing weights of the network.

Recent work in the context of knowledge consolidation has proposed to rehearse previous tasks whilst learning new tasks. Pseudo-recursal (Atkinson et al., 2018a) used Generative Adversarial Networks (GAN) (Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville, & Bengio, 2014) to generate examples from previous tasks used to retrain old tasks while learning a new task. Similar to continual learning research described previously, research in this area has mainly focused on avoiding *catastrophic forgetting* of previous tasks. Silver et al. (2015) also investigated the problem of task rehearsal. Similar to Pseudo-recursal, the authors proposed to generate virtual training examples of previous tasks based on knowledge of the probability distributions of the inputs of these tasks. The authors also pointed to the possibility of improving previous knowledge while learning new tasks.

6.2 Knowledge Collected while Transferring Forward

The method presented in section 5 proposed to transfer selected support vectors from a set of source hypotheses S to a target task with the aim to aid learning of a target function or hypothesis $f^{(t)}$. As a result of this transfer, tuples of the form:

$$Z = [(x^{(s)}, y^{(s)}, \alpha^{(s)}), (x^{(t)}, y^{(t)}, \alpha^{(t)})] \tag{21}$$

that match a particular source support vector $(x^{(s)}, y^{(s)}, \alpha^{(s)})$ with a related target support vector $(x^{(t)}, y^{(t)}, \alpha^{(t)})$ learned for $f^{(t)}$, which were involved in transfer while transferring forward using Eq. 16, can be identified. A pair of this kind can be understood as a subspace of knowledge that is shared by $f^{(s)}$ and $f^{(t)}$. Therefore, exploiting these tuples

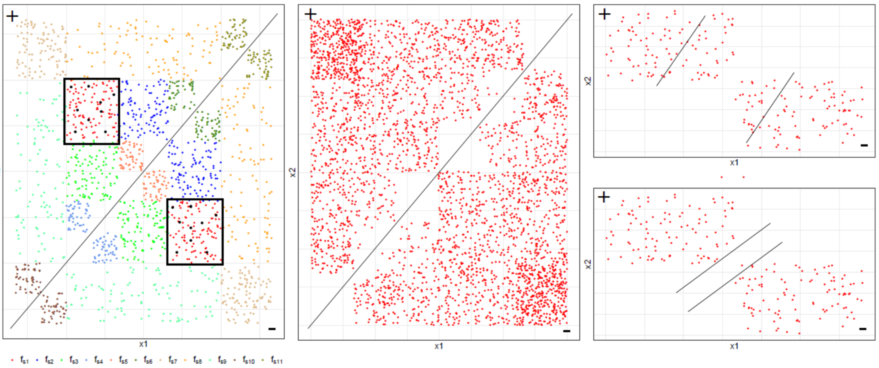


Figure 6: An abstract example, with: training examples for a set of hypotheses $\{f_1^{(s)}, f_2^{(s)}, \dots, f_{11}^{(s)}\}$ (left), where $f_1^{(s)}$ is shown in the black squares, a test set (center), and two extreme scenarios when refining an existing $f_1^{(s)}$ (right-top no refinement, right-bottom no retention).

could be potentially useful for a learning system that aims to refine existing $f^{(s)}$. The next section describes a method that pursues this goal.

6.3 The Problem of Hypothesis Refinement with Retention

We start with an abstract example of the proposed approach, represented in Figure 6 (left). Training examples for eleven two-dimensional binary classification tasks are shown in colors. For each problem, examples above the diagonal line are labelled positive, and examples below this line are labelled negative. The aim is to learn an optimal separating hyperplane for each problem. Suppose we learn $f_1^{(s)}$ (shown in the black squares in Figure 6) first. An initial SVM hypothesis $f_1^{(s)}$ will be suboptimal due to the limitation of the training examples, and because of noise in some of these examples (black dots inside the squares, that correspond to examples with shuffled labels). Since separating hyperplanes of $f_2^{(s)}$ to $f_{11}^{(s)}$ are similar to $f_1^{(s)}$, refining $f_1^{(s)}$ by transferring (backward) from $\{f_2^{(s)}, \dots, f_{11}^{(s)}\}$ while these are learned sequentially may help to improve the performance of $f_1^{(s)}$ on unobserved examples (center).

Now consider the two extreme scenarios for an $f^{(s)}$, as shown in Figure 6 (right). Suppose that the hypothesis of interest, $f_1^{(s)}$, changes as a result of the refinement process. The first scenario (top) shows no refinement: moving $f_1^{(s)}$ produces a wider margin with potentially worse performance on an unobserved sample. The second scenario (bottom) shows no retention: moving $f_1^{(s)}$ produces an extremely narrow margin and almost no training examples will be retained as support vectors.

6.4 Hypothesis Refinement with Retention using ν-SVM

We first define hypothesis refinement with retention using selective knowledge transfer.

Definition 4. ***Hypothesis refinement by selective knowledge transfer with retention*** *Given a source hypothesis $f^{(s)}$ and a hypothesis or function $f^{(t)}$ learned on a recent* *target task, with $f^{(t)} = \{(x_1^{(t)}, y_1^{(t)}, \alpha_1^{(t)}), \ldots, (x_l^{(t)}, y_l^{(t)}, \alpha_l^{(t)})\}$ generated using the method described in Eq. 16, hypothesis refinement with retention of knowledge by means of selective knowledge transfer aims to use knowledge collected in the form of tuples such as in Eq. 21, $Z = [(x^{(s)}, y^{(s)}, \alpha^{(s)}), (x^{(t)}, y^{(t)}, \alpha^{(t)})]$, to obtain a refined version of $f^{(s)}$, $f^{(s*)}$, that has potentially better performance on unobserved samples. The refined version satisfies $l^* \not\ll l$, i.e. the number l^* of support vectors in the refined version $f^{(s*)}$ is not much smaller than the number l of support vectors in $f^{(s)}$.*

We propose **HRSVM**, a method to transfer selected knowledge backward from a target hypothesis learned recently to related source hypotheses. The aim is to refine an existing $f^{(s)}$ using knowledge collected during a target task that is executed after these sources were learned. We use **AccGenSVM**, presented in Section 5, to select these support vectors. This strategy emphasizes refinement towards the knowledge shared between the source and the recent target, particularly when knowledge sharing occurs across different subspaces of these hypotheses. This composes the selective nature of the proposed transfer backward method.

A local function or hypothesis that uses one of these tuples as training examples is learned as a representation of a subspace of shared knowledge. Such function acts as the kind of intermediate knowledge described by Jonschkowski et al. (2015), which in our case is aimed to be useful for refining the existing hypothesis. This knowledge is stored only temporarily during refinement. We propose to learn as many of these functions as tuples can be collected while transferring to a target task.

To accomplish the goal of knowledge retention while refining, we base our **HRSVM** solution on ν-SVM (Schölkopf et al., 2000). ν-SVM (Schölkopf et al., 2000) was proposed as an alternative to the C-SVM problem for classification proposed by Vapnik (1998). The parameter ν has three properties: 1) it acts as an upper bound on the fraction of margin errors, 2) it acts as a lower bound on the fraction of support vectors, 3) for i.i.d samples and with analytic and non-constant kernels, ν equals both the fraction of support vectors and the fraction of errors. In our case, the desired effect is to control the margin size by: 1) disencouraging compression, *i.e.* maximising the number of examples selected as support vectors in the refined hypothesis, 2) refining appropriately on the training (support vectors) set, *i.e.* minimising the training error. The need to control for the margin size whilst transferring was pointed out by Aytar et al. (2011), for cases of transfer to a target task. We formulate a modified ν-SVM problem that regularizes the distance between the parameter vector $\mathbf{w}$ to be learned and a parameter vector derived from representations of subspaces of shared knowledge, $\mathbf{w^{(d)}}$. The aim is to obtain a solution on the hypothesis space that is biased towards the knowledge shared with the target.

Our (soft-margin) ν-SVM problem is:

$$\begin{aligned} &\min_{\mathbf{w},\boldsymbol{\xi},\rho} \frac{1}{2}\|\mathbf{w} - \Gamma\mathbf{w^{(d)}}\|_2^2 - \nu\rho + \frac{1}{l}\sum_{i=1}^{l}\xi_i \\ &s.t. \begin{cases} \forall i \quad y_i(\mathbf{w}^\top\phi(x_i) + b) \geq \rho - \xi_i \\ \forall i \quad \xi_i \geq 0, \rho \geq 0 \end{cases} \end{aligned} \tag{22}$$

which maximises for l support vectors on the source hypothesis, $(x_i, y_i), 0 \leq i \leq l$, by learning a weight vector $\mathbf{w}$ of the function separating these support vectors. A set of slack variables $\boldsymbol{\xi} = \{\xi_1,...,\xi_n\}$ allows some of these examples to violate the margin constraints, whilst ρ is a learned parameter. Note that for $\boldsymbol{\xi} = 0$, *i.e.* a hard margin, the first constraint implies that the two classes are separated by the margin $2\rho/||\mathbf{w}||$. Therefore, both $\mathbf{w}$ and ρ define the margin size. The proposed problem is inbetween regularizing fully between a target and a source such as in A-SVM (Yang et al., 2007), and learning with privileged information using SVM+ (Wang & Hebert, 2016) which learns these two sets in parallel by regularizing their sum. We fix the parameter vector $\mathbf{w}^{(\mathbf{d})}$ while optimizing Eq. 22, since it is derived before solving (minimising) this problem, as will be explained in the dual formulation. In practice $\mathbf{w}^{(\mathbf{d})}$ are multiple sets of parameters, one for each subspace of shared knowledge. The influence of $\mathbf{w}^{(\mathbf{d})}$ is controlled by Γ. This value can be set experimentally by, for example, grid-search over a range of values.

Based on (Schölkopf et al., 2000), the Lagrangian representation of the HRSVM primal problem is:

$$\begin{aligned}\min_{\mathbf{w},\boldsymbol{\xi},\rho}\max_{\boldsymbol{\alpha},\boldsymbol{\beta},\delta}\mathscr{L}(\mathbf{w},\boldsymbol{\xi},\rho,\boldsymbol{\alpha},\boldsymbol{\beta},\delta) = & \frac{(\mathbf{w}-\boldsymbol{\Gamma}\mathbf{w}^{(\mathbf{d})})^\top(\mathbf{w}-\boldsymbol{\Gamma}\mathbf{w}^{(\mathbf{d})})}{2} \\ & -\nu\rho + \frac{1}{l}\sum_{i=1}^{l}\xi_i \\ & +\sum_{i=1}^{l}\alpha_i\{1-y_i(w^\top\phi(x_i)+b)-\rho+\xi_i)\} \\ & -\sum_{i=1}^{l}\beta_i\xi_i - \delta\rho\end{aligned} \tag{23}$$

The dual objective of **HRSVM** is therefore:

$$\begin{aligned}&\max_{\boldsymbol{\alpha}} -\frac{1}{2}\sum_{i,j=1}^{l,l}\alpha_i\alpha_j y_i y_j K(x_i,x_j) - \Gamma\sum_{i,k=1}^{l,2m}\alpha_i y_i \alpha_k^{(d)} y_k^{(d)} K(x_k^{(d)},x_i) \\ &s.t.\ \forall i\ \ 0 \leq \alpha_i \leq 1/l, \sum_{i=1}^{l} y_i\alpha_i = 0, \sum_{i=1}^{l}\alpha_i \geq \nu\end{aligned} \tag{24}$$

with $\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j K(x_i,x_j)$ the space of l source support vectors in the current $f^{(s)}$ and $\sum_{i,k=1}^{l,2m}\alpha_i y_i \alpha_k^{(d)} y_k^{(d)} K(x_k^{(d)},x_i)$ the representations of subspaces of shared knowledge between source support vectors in $f^{(s)}$ and target support vectors in $f^{(t)}$. Here, each $(\alpha_k^{(d)}, y_k^{(d)}, x_k^{(d)})$, with $1 \leq k \leq 2m$, are terms extracted from m functions $f^{(d)}$ learned with one-class SVM (Schölkopf, Platt, Shawe-Taylor, Smola, & Williamson, 2001). Each of these functions uses one tuple of the form in Eq. 21, $Z = \{(x^{(s)}, y^{(s)}, \alpha^{(s)}), (x^{(t)}, y^{(t)}, \alpha^{(t)})\}$ as training examples. Although other mechanisms may be applicable, one-class SVM makes the inclusion of this intermediate representation into the final objective straightforward. In binary classification tasks $Z = \{(x^{(s)}, y^{(s)}, \alpha^{(s)}), (x^{(t)}, y^{(t)}, \alpha^{(t)})\}$ should be such that $y^{(s)} = 1$ and $y^{(t)} = 1$ or $y^{(s)} = -1$ and $y^{(t)} = -1$, for transfer to occur among corresponding classes.

Chen et al. (2005) derived the maximal feasible value of ν as $\nu = 2 * min(l_+, l_-)/l$, where l_+ and l_- correspond to the number of positive and negative examples, respectively.

The maximal feasible value should be enforced, such that the maximal knowledge (number of source support vectors) is retained, whilst the second term in Eq. 24 drives learning of the refined $f^{(s)}$ towards the subspaces of shared knowledge. In the sequential learning setting with **HRSVM**, ρ of the refined hypothesis (in Eq. 22) should be encouraged to be larger than the corresponding parameter of the current hypothesis.

Similar to **AccGenSVM** presented in Section 5, the proposed method can be understood as an application of the idea of learning with privileged information (Vapnik & Izmailov, 2016). In particular, **HRSVM** is connected to the multi-view (correlation) pattern for learning with privileged or side information (Jonschkowski et al., 2015). This pattern implies that intermediate representations computed from related examples are useful for predicting the target output. Our proposed solution could be compared to existing hypothesis transfer (forward) methods such as A-SVM (Yang et al., 2007). However, this general strategy has been shown to underperform the selective transfer (forward) setting presented in Section 5 in scenarios of small training samples. Our strategy could be also compared to existing lifelong learning methods such as ELLA (Ruvolo & Eaton, 2013b), where shared knowledge is identified and updated to encourage refinement. However, rather than learning a global layer of shared knowledge across all tasks, we hypothesise that exploiting subspaces of shared knowledge can help to refine the existing knowledge more selectively, similar to the transfer forward setting presented earlier in this paper. Naturally, performing this selective refinement requires a mechanism to select which source hypotheses and which subspaces of these source hypotheses to refine. In our case, both of these rely on the selection performed while transferring forward. For selecting which sources, the KL-divergence is used to determine relatedness of a source hypothesis and target training examples. For selecting subspaces in these sources and the target recently learned, a simple k-nearest neighbours mechanism is used between source support vectors and target training examples. As opposed to methods such as CL (Fei et al., 2016), and similar to ELLA, **HRSVM** relies on source and target hypotheses rather than on source or target data to perform transfer. As opposed to existing continual learning methods such as DEN (Yoon et al., 2017a), **HRSVM** aims to retain knowledge whilst refining hypotheses, rather than to avoid catastrophic forgetting of existing knowledge.

Algorithm 2 describes the proposed approach. For completeness, we also include **AccGenSVM**. Similar to **AccGenSVM**, **HRSVM** uses SMO (Bottou & Lin, 2007) as solver. Similar to **AccGenSVM**, **HRSVM** operates in two phases:

Phase 1 - Learning of representations of subspaces of shared knowledge.

1. For a particular source hypothesis $f^{(s)}$, select tuples Z represented as in Eq. 21 such that for an $x^{(s)} \in Z$ also $x^{(s)} \in f^{(s)}$.

2. Learn a one-class SVM function $f^{(d)}$ using examples in Z recovered in the previous step as training examples.

3. Learn as many $f^{(d)}$ as $x^{(s)} \in f^{(s)}$ such that $x^{(s)} \in Z$, for every Z stored as a tuple of the form in Eq. 21. Accumulate these $f^{(d)}$ into a set of M of these functions, to be used in the next phase.

Phase 2 - Refinement of a hypothesis.

1. Solve the problem in Eq. 24 using the current l support vectors of the current $f^{(s)}$ and the M representations learned in Phase 1.

6.5 Theoretical and Complexity Properties of HRSVM

We use the VC-dimension as a framework to analyse the theoretical properties of HRSVM. We start by recalling the well-known VC-dimension bounds for SVM classifiers (Vapnik, 1998):

Theorem 1. *Let vectors $x \in X$ belong to a sphere of radius R. Then the set of Δ-margin separating hyperplanes has VC-dimension $VC\text{-}dim(\mathbb{H})$ bounded by:*

$$VC\text{-}dim(\mathbb{H}) \leq min\left(\frac{R^2}{\Delta^2}, d\right) + 1 \tag{25}$$

where d is the dimensionality of the problem, R the radius of a sphere enclosing the training examples and Δ the size of the margin. For sufficiently large d, and since $R = 1$ without loss of generality, the only way to alter these bounds is by changing the margin Δ.

We derive VC-dimension bounds of the proposed **HRSVM** algorithm inspired by the research of Abu-Mostafa (1995). This framework studies the use of *hints* or additional information to encourage higher generalisation of a chosen function $f^{(s)}$. These hints can be, for example, invariances of the training examples to transformations such as rotations, translations, etc. The knowledge we propose to collect as $Z = \{(x^{(s)}, y^{(s)}, \alpha^{(s)}), (x^{(t)}, y^{(t)}, \alpha^{(t)})\}$, is an example of such hints.

First, let $D = \bigcup_c D_c$ be the set of examples D partitioned into D_c classes. In our case, each D_c is composed of one or more x examples and their corresponding $x^{(d)}$, such that $D_c = \{(x_1, x^{(d)}), \ldots, (x_l, x^{(d)}) \in Z\}$, with Z in Eq. 21. The value of the desired function $f^{(s)}$ is constant in each of these classes (Abu-Mostafa, 1995), *i.e.* for all of the $(x_i, x_i^{(d)}) \in D_c$ the value of a learned $f^{(s)}$ is more likely to be constant. When D_c contains a single $(x, x^{(d)})$ then $f^{(s)}(x) \simeq f^{(d)}(x^{(d)})$, *i.e.* D_c is only marginally useful since it chooses an $f^{(s)}$ that is likely to be constant on x only, given $x^{(d)}$. The extent to which this is useful depends on the *quality* of $x^{(d)}$ for the given task. At the other extreme, if a given D_c contains all pairs $(x, x^{(d)})$ then this hint is extremely useful since it denotes that $f^{(s)}$ is constant at D, with $f^{(s)} = 1$ or $f^{(s)} = -1$. In more realistic cases where a given D_c contains some $(x, x^{(d)})$, an $f^{(s)}$ to be learned will be more likely to be constant at these x points, with $f^{(s)}(x) = 1$ or $f^{(s)}(x) = -1$. Therefore the higher the chance of learning a decision boundary that fits these examples well. These x will have a higher chance of becoming margin support vectors with $0 < \alpha < C$.

Abu-Mostafa (1995) demonstrated that the VC-dimension of a hypotheses set $\mathbb{H}$ conditioned on a hypotheses set $\mathbb{G}$ given by hints $D = \bigcup_c D_c$, such as for example invariance hints, is such that $VC\text{-}dim(\mathbb{H}|\mathbb{G}) \leq VC\text{-}dim(\mathbb{H})$, as follows:

Algorithm 2: Pseudo-code for the **HRSVM** algorithm for hypothesis refinement. T only retains Z such that $x^{(t)}$ is selected as a support vector of $f^{(t)}$.

```
Data: Target data $D^{(t)}$, source hypotheses set $S$, $\tau$ threshold for KL-divergence,
      $x_1^{(t)}, x_2^{(t)} \leftarrow$ points selected by SMO on the target task $T^{(t)}$, $c^{(t)} \leftarrow 0, F \leftarrow \emptyset$,
      $T \leftarrow \emptyset$
Result: Updated source hypotheses subset $F$
// Transfer forward to $T^{(t)}$ using AccGenSVM
while Eq. 16 not solved do
    forall $f^{(s)}$ in $S$ do
        if $f^{(s)}$ related to $D^{(t)}$ (according to KL-divergence) then
            $c^{(t)} \leftarrow 0$
            forall $x^{(s)}$ in $f^{(s)}$ do
                // $x^{(t)} = x_1^{(t)}$ or $x^{(t)} = x_2^{(t)}$
                if $x^{(s)}$ is a k-nearest neighbour of $x^{(t)}$ and $y^{(s)} \times y^{(t)} == 1$ then
                    $c^{(t)} \leftarrow c^{(t)} + \alpha^{(s)}$
                    $Z \leftarrow \{(x^{(s)},y^{(s)},\alpha^{(s)}), (x^{(t)},y^{(t)},\alpha^{(t)})\}$
                    $T \leftarrow \{T, (f^{(s)}, Z)\}$
                end
            end
            $F \leftarrow \{F, f^{(s)}\}$
        end
    end
    $c^{(t)} = c^{(t)} \times |F|/|S|$
end
// Transfer backward to hypotheses $F$ (proposed HRSVM)
forall $f^{(s)}$ in $F$ do
    $M \leftarrow \emptyset$
    forall $(f, \{(x^{(s)},y^{(s)},\alpha^{(s)}), (x^{(t)},y^{(t)},\alpha^{(t)})\})$ in $T$ do
        if $f == f^{(s)}$ then
            // Phase 1
            $f^{(d)} \leftarrow$ one-class$(x^{(s)}, x^{(t)})$
            $M \leftarrow \{M, f^{(d)}\}$
        end
    end
    if $M$ not $\emptyset$ then
        // Phase 2
        $f^{(s)*} \leftarrow$ solve$(f^{(s)}, M)$ using Eq. 24
        $f^{(s)} \leftarrow f^{(s)*}$
    end
end
```

Theorem 2. *Let $VC\text{-}dim(\mathbb{H}|\mathbb{G})$ be the VC-dimension of a set of hypotheses $\mathbb{H}$ conditioned on a set of hypotheses $\mathbb{G}$ given by $D = \bigcup_c D_c$, where for each $\hbar \in \mathbb{H}$ either $\hbar$ satisfies $\mathbb{G}$ or does not satisfy it, i.e. for all $(x, x^{(d)}) \in D_c$ then $\hbar(x)$ is constant. Since the set of hypotheses that satisfies $\mathbb{G}$ is $\hat{\mathbb{H}}$:*

$$\hat{\mathbb{H}} = \{\hbar \in \mathbb{H} | \textit{for all pairs } (x, x^{(d)}) \in D_c \textit{ then } \hbar(x) \textit{ is constant}\} \tag{26}$$

Then:

$$VC\text{-}dim(\mathbb{H}|\mathbb{G}) \leq VC\text{-}dim(\mathbb{H}) \tag{27}$$

Proof. Since $\hat{\mathbb{H}} \subseteq \mathbb{H}$, then $VC\text{-}dim(\mathbb{H}|\mathbb{G}) \leq VC\text{-}dim(\mathbb{H})$. □

Nontrivial hints such as invariance hints lead to a substantial reduction from $\mathbb{H}$ to $\hat{\mathbb{H}}$, and as a result potentially $VC\text{-}dim(\mathbb{H}|\mathbb{G}) < VC\text{-}dim(\mathbb{H})$ (Abu-Mostafa, 1995).

We now derive specific bounds encouraged by our SVM hypothesis refinement method as follows:

Theorem 3. *Let $D = \bigcup_c D_c$ be the training examples D partitioned into D_c classes, where each D_c is composed of one or more x examples and a corresponding $x^{(d)}$, such that $D_c = \{(x_1, x_1^{(d)}), \ldots, (x_l, x_l^{(d)})\}$ where both $\{x_1, \ldots, x_l\}$ and $\{x_1^{(d)}, \ldots, x_l^{(d)}\} \in Z$, with Z in Eq. 21. Let $f^{(s*)}$ be a function learned using Eq. 24, such that $f^{(s*)}$ is desired to be constant for a given D_c. Let $\mathbb{H}$, $\mathbb{G}$, $VC\text{-}dim(\mathbb{H})$ and $VC\text{-}dim(\mathbb{H}|\mathbb{G})$ as defined in Theorem 2. Let $\mathbf{w}^*$ be the weight vector learned for $f^{(s*)}$. Let $\Delta^* = 2/||\mathbf{w}^*||$ be the corresponding margin. Let $\mathbf{w}$ be the weight vector and $\Delta = 2/||\mathbf{w}||$ the margin of the existing $f^{(s)}$, learned using a regular SVM (Vapnik, 1998) or **HRSVM**. Since:*

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \phi(x_i) \tag{28}$$

and:

$$\mathbf{w}^* = \sum_{i=1}^{l^*} \alpha_i^* y_i \phi(x_i) \tag{29}$$

and:

$$l^* \leq l \tag{30}$$

For:

$$\begin{gathered} \boldsymbol{\alpha_m} = \{\alpha | 0 < \alpha < C\}, \boldsymbol{\alpha_b} = \{\alpha | \alpha = C\}, \\ \boldsymbol{\alpha_0} = \{\alpha | \alpha = 0\} \end{gathered} \tag{31}$$

and:

$$\begin{gathered} \boldsymbol{\alpha^*_m} = \{\alpha^* | 0 < \alpha^* < C\}, \boldsymbol{\alpha^*_b} = \{\alpha^* | \alpha^* = C\}, \\ \boldsymbol{\alpha^*_0} = \{\alpha^* | \alpha^* = 0\} \end{gathered} \tag{32}$$

With:

$$|\boldsymbol{\alpha^*_m}| \geq |\boldsymbol{\alpha_m}| \tag{33}$$

Then:

$$\Delta^* \geq \Delta \tag{34}$$

and, by Theorem 3 in Vapnik (1998) :

$$VC\text{-}dim(\mathbb{H}|\mathbb{G}) \leq VC\text{-}dim(\mathbb{H}) \tag{35}$$

Proof. Since by $D = \bigcup_c D_c$, $f^{(s*)}$ is more likely to be constant on several $x \in D_c$, more α^* will be forced to have values between 0 and C, *i.e.* to lie exactly on the margin. Therefore, with $l^* \leq l$, then $||\mathbf{w}^*|| \leq ||\mathbf{w}||$. Since $\Delta^* = 2/||\mathbf{w}^*||$ and $\Delta = 2/||\mathbf{w}||$, then necessarily $\Delta^* \geq \Delta$. Furthermore, since from Eq. 7.15 in Schölkopf et al. (2000):

$$\rho = \frac{1}{2s}\left(\sum_{x \in S_+}\sum_j \alpha_j y_j K(x, x_j) - \sum_{x \in S_-}\sum_j \alpha_j y_j K(x, x_j)\right) \tag{36}$$

with S_+ the set of positive examples with $0 < \alpha < 1$ and S_- the set of negative examples with $0 < \alpha < 1$, where $|S_+| = |S_-| = s$. By requiring that $\rho^* \geq \rho$ necessarily the set $\{S_+, S_-\}$ has to be bigger (*i.e.* more training examples are support vectors lying exactly on the margin) or their difference is larger. In the first case, $||w^*|| < ||w||$, and with $\rho^* \geq \rho$ then $\Delta^* > \Delta$. In the second case, at least $||w^*|| = ||w||$, and therefore at least $\Delta^* = \Delta$. □

The extent to which $D = \bigcup_c D_c$ will be beneficial for refining an $f^{(s)}$ will be driven by how appropriate are tuples in Eq. 21. This will highly depend on the relatedness of the underlying distributions of the source hypothesis $f^{(s)}$ and the target hypothesis $f^{(t)}$. Below we analyse the effects of relatedness of tasks, and link these results to the bound just presented. We will demonstrate that a smaller VC-dimension is key to achieve the error bounds that can be satisfied by transferring knowledge across related tasks.

6.5.1 Relatedness of Tasks

We adopt the framework by Ben-David and Borbely (2008) for studying the effects of relatedness of tasks in our proposed hypothesis refinement method. This framework focuses on analysing improvement on the error bounds of a particular task when it is learned in parallel with other tasks, *i.e.* when a task is learned using multitask learning. This framework is an extension of preceding research which focused on analysing these effects for the group of tasks (Baxter et al., 2000). Ben-David and Borbely (2008) defined task relatedness in terms of relatedness of the distributions underlying these tasks, as follows:

Definition 5. *Let $\{P^{(1)}, \ldots, P^{(N)}\}$ be the underlying probability distributions of a set of tasks N over a domain X. Let $\mathcal{F}$ be a set of transformations $f : X \to X$. Let $P^{(1)}$ and $P^{(2)}$ be related if one can be generated from the other by applying some $f \in \mathcal{F}$, such that $P^{(1)} = f[P^{(2)}]$ or $P^{(2)} = f[P^{(1)}]$. The samples $\{D^{(1)}, \ldots, D^{(T)}\}$ to be used during the learning tasks $\{T^{(1)}, \ldots, T^{(N)}\}$ are said to be $\mathcal{F}$-related if these samples come from $\mathcal{F}$-related probability distributions.*

Let $\mathbb{H}$ be a hypothesis space over that domain, and $\mathbb{H}$ be closed under the action of $\mathcal{F}$. Let $\mathcal{H}$ be a family of hypothesis spaces that consist of sets of hypotheses in $\mathbb{H}$ which are equivalent up to transformations in $\mathcal{F}$. If $\mathcal{F}$ acts as a group over $\mathbb{H}$ because:

- *For every $f \in \mathcal{F}$ and every $h \in \mathbb{H}$, $h \circ f \in \mathbb{H}$, and*
- *$\mathcal{F}$ is closed under transformation composition and inverses, i.e. for every $f, g \in \mathcal{F}$, the inverse transformation, f^{-1}, and the composition, $f \circ g$ are also members of $\mathscr{F}$*

Then the equivalence relation $\sim_{\mathcal{F}}$ on $\mathbb{H}$ is defined by: $h^{(1)} \sim_{\mathcal{F}} h^{(2)} \iff$ there exists $f \in \mathcal{F}$ such that $h^{(2)} = h^{(1)} \circ f$.

Therefore the framework considers the family of hypothesis spaces, $\mathcal{H} = \{[h] : [h \in \mathbb{H}]\}$, which is the family of all equivalence classes of $\mathbb{H}$ under $\sim_{\mathcal{F}}$ ($\mathcal{H} = \mathbb{H}/\sim_{\mathcal{F}}$).

Therefore the framework considers the family of hypothesis spaces, $\mathcal{H} = \{[h] : [h \in \mathbb{H}]\}$, which is the family of all equivalence classes of $\mathbb{H}$ under $\sim_{\mathcal{F}}$ ($\mathcal{H} = \mathbb{H}/\sim_{\mathcal{F}}$).

The original setting of this framework is in multitask learning where the equivalence class $[h]$ is first found using samples from all tasks. This requires to first identify aspects of all tasks that are invariant under $\mathcal{F}$. A second step restricts the learning of a particular task $T^{(i)}$ using a sample $D^{(i)}$ to select a hypothesis $h' \in [h]$ as the hypothesis for that task. The benefit is a smaller hypothesis space to be explored, since the original $\mathbb{H}$ is replaced by a subset $[h]$ that represents these invariances, provided these exist.

Our hypothesis refinement problem is a special case of this framework, which is limited to two tasks at a time: a source $T^{(s)}$ and a target $T^{(t)}$ task. We aim to improve performance on one of these tasks, $T^{(s)}$. Although these tasks are not learned in parallel such as in the multitask learning setting, our method proposes to share knowledge that is common to both tasks, as explained in Section 6. We transfer knowledge from the space of a target probability distribution $P^{(t)}$, represented by a sample $D^{(t)}$ which has been recently used to learn a hypothesis $f^{(t)}$ by solving Eq. 16, to the space of a hypothesis $f^{(s)}$ that was learned on a probability distribution $P^{(s)}$ using a sample $D^{(s)}$ (or a previous version of $f^{(s)}$ learned on that sample). Although we do not learn $T^{(t)}$ and $T^{(s)}$ in parallel, similar to multitask learning our aim is to find transformations from $P^{(t)}$ to $P^{(s)}$ using samples $D^{(t)}$ and $D^{(s)}$ to bias learning of a refined version of $f^{(s)}$ towards aspects that are invariant with an $f^{(t)}$ learned recently. We are interested in analysing the effect of this transfer process in the performance of the source hypothesis $f^{(s)}$.

Similar to other learning problems, the aim of our hypothesis refinement method is to find a hypothesis $h \in \mathbb{H}$ that is optimal according to the empirical error on a set of examples D, $\hat{\mathbb{E}}r^{D}(h)$, and that approximates well the true error $\mathbb{E}r^{P}$. The latter is evaluated by the performance of the best hypothesis on the space of that task, by $\mathbb{E}r^{P}(\mathbb{H}) = \inf_{h \in \mathbb{H}} \mathbb{E}r^{P}(h)$.

We define the aim of our framework similar to Ben-David and Borbely (2008). For our special case of two tasks, one source $T^{(s)}$ and one target $T^{(t)}$:

Definition 6. *Given classes $\mathcal{F}$ and $\mathbb{H}$, and a pair of labeled samples $D^{(s)}$, $D^{(t)}$ for tasks $T^{(s)}$, $T^{(t)}$, the proposed method:*

- *Selects $h^* \in \mathbb{H}$ that minimises $\inf_{h_1,\ldots,h_n \in [h]} \left(\hat{\mathbb{E}}r^{D^{(s)}}(h) + \Gamma \hat{\mathbb{E}}r^{D^{(t)}}(h)\right)$ over all $[h] \in \mathbb{H}/\sim_{\mathcal{F}}$, with $\Gamma \in [0,1]$ usually small.*
- *Selects $h^{\diamondsuit} \in [h^*]$ that minimises $\mathbb{E}r^{D^{(s)}}(h')$ over all $h' \in [h^*]$, and outputs $h^{\diamondsuit}$ as the hypothesis for task $T^{(s)}$.*

The proposed algorithm executes these two steps at the same time while optimizing for the problem in Eq. 24. Also, note that in the infimum $\inf_{\hslash_1,\ldots,\hslash_n\in[\hslash]}\big(\hat{\mathbb{E}}r^{D^{(s)}}(\hslash)+\Gamma\hat{\mathbb{E}}r^{D^{(t)}}(\hslash)\big)$ the term $\hat{\mathbb{E}}r^{D^{(s)}}(\hslash)$ is given more relevance since the aim is to find the hypothesis $\hslash\in[\hslash]$ that performs better for the source task according to the empirical risk $\hat{\mathbb{E}}r^{D^{(s)}}$.

Based on this definition, and similar to Theorem 3 in Ben-David and Borbely (2008), in our special case of two tasks $T^{(s)}$ and $T^{(t)}$.

Theorem 4. *Let $P^{(s)}$ and $P^{(t)}$ be a set of $\mathcal{F}$-related probability distributions, $D^{(s)}$ and $D^{(t)}$ random samples representing these distributions on tasks $T^{(s)}$ and $T^{(t)}$ respectively, $\mathcal{F}$ a family of domain transformations of the domain X and let $\mathbb{H}$ be a family of binary valued functions on that domain such that $\mathcal{F}$ acts as a group over $\mathbb{H}$. Let $d_{max}=max_{\hslash\in\mathbb{H}}VC\text{-}dim([\hslash]_{\mathcal{F}})$. Let $\hslash^{\Diamond}$ be selected according to Definition 6. Then, for every ϵ_1, ϵ_2, $\delta>0$, if:*

$$|D^{(s)}|\geq\frac{64}{\epsilon_1^2}\left[2d_{max}log\frac{12}{\epsilon_1}+log\frac{8}{\delta}\right] \tag{37}$$

and:

$$|D^{(t)}|\geq\frac{88}{\epsilon_2^2}\left[2d_{\mathcal{H}}(2)log\frac{22}{\epsilon_2}+\frac{1}{2}log\frac{8}{\delta}\right] \tag{38}$$

then with probability greater than $(1-\delta)$:

$$\mathbb{E}r^{P^{(s)}}(\hslash^{\Diamond})\leq\inf_{\hslash\in\mathbb{H}}\mathbb{E}r^{P^{(s)}}(\hslash)+2(\epsilon_1+\epsilon_2) \tag{39}$$

Proof. Let $\hslash^{\#}$ be the best $P^{(s)}$ label predictor in $\mathbb{H}$, *i.e.* $\hslash^{\#}=\arg\min_{\hslash\in\mathbb{H}}\mathbb{E}r^{P^{(s)}}(h)$. Let $[\hslash^{*}]$ be the equivalence class picked according to Definition 6. By the choice of $\hslash^{*}$:

$$\inf_{\hslash_1,\ldots,\hslash_n\in[\hslash^{*}]}\big(\hat{\mathbb{E}}r^{D^{(s)}}(\hslash)+\Gamma\hat{\mathbb{E}}r^{D^{(t)}}(\hslash)\big)\leq\inf_{\hslash_1,\ldots,\hslash_n\in[\hslash^{\#}]}\big(\hat{\mathbb{E}}r^{D^{(s)}}(\hslash)+\Gamma\hat{\mathbb{E}}r^{D^{(t)}}(\hslash)\big) \tag{40}$$

By Theorem 2 in Ben-David and Borbely (2008), in our case of two tasks:

$$\left|\mathbb{E}r^{P^{(s)}}([\hslash])-\inf_{h_1,\ldots,h_n\in[\hslash]}\frac{1}{2}(\hat{\mathbb{E}}r^{D^{(s)}}(\hslash)+\Gamma\hat{\mathbb{E}}r^{D^{(t)}}(\hslash))\right|\leq\epsilon_1 \tag{41}$$

then with probability greater than $(1-\delta/2)$:

$$\inf_{\hslash_1,\ldots,\hslash_n\in[\hslash^{\#}]}\big(\hat{\mathbb{E}}r^{D^{(s)}}(\hslash)+\Gamma\hat{\mathbb{E}}r^{D^{(t)}}(\hslash)\big)\leq\mathbb{E}r^{P^{(s)}}([\hslash^{\#}])+\epsilon_1 \tag{42}$$

and:

$$\mathbb{E}r^{P^{(s)}}([\hslash^{*}])\leq\inf_{\hslash_1,\ldots,\hslash_n\in[\hslash^{*}]}(\hat{\mathbb{E}}r^{D^{(s)}}(\hslash)+\Gamma\hat{\mathbb{E}}r^{D^{(t)}}(\hslash))+\epsilon_1 \tag{43}$$

Then, combining the inequalities above, with probability greater than $(1-\delta/2)$:

$$\mathbb{E}r^{P^{(s)}}([\hslash^{*}])\leq\mathbb{E}r^{P^{(s)}}([\hslash^{\#}])+2\epsilon_1 \tag{44}$$

Since $\hslash^{\Diamond}\in\hslash^{*}$, with probability greater than $(1-\delta/2)$, $\hslash^{\Diamond}$ will have an error for $P^{(s)}$ which is within $2\epsilon_2$ of the best hypothesis there, *i.e.* $\mathbb{E}r^{P_s}([\hslash^{*}])$. □

These bounds depend on the gap between three parameters: d_{max}, $d_{\mathcal{H}}(N)$, with N the number of tasks, and $VC\text{-}dim(\mathbb{H})$. In the general setting of Ben-David and Borbely (2008), the first and second parameters depend on the VC-dimension of the new space of hypotheses $[\hbar]$, with both $d_{max} = max_{\hbar \in \mathbb{H}} VC\text{-}dim([\hbar]_{\sim \mathcal{F}})$ and $d_{\mathcal{H}}(N) = \max_{\hbar \in \mathbb{H}} VC\text{-}dim([\hbar]_{\sim \mathcal{F}})$, with $\mathcal{H}$ in Definition 5. As explained by Theorem 3, the proposed method encourages a larger margin, a fact that is associated to a smaller VC-dimension. This benefits the bounds demonstrated in Theorem 4.

6.5.2 Unrelated Tasks

For non-related tasks the set of transformations $\mathcal{F}$ may not even exist. Therefore, these bounds are not expected to change. However, negative transfer may occur if transfer is encouraged regardless of $\mathcal{F}$ not existing. In practice, we control this risk by determining the relatedness of tasks before transferring. This relatedness is measured using KL-divergence, as explained in Section 5. To recall, the KL-divergence from a distribution $P^{(s)}$ to a distribution $P^{(t)}$ measures the amount of information lost when $P^{(s)}$ is used to approximate $P^{(t)}$. KL-divergence is formulated as:

$$KL(P^{(t)}||P^{(s)}) = P^{(t)} \log \left(\frac{P^{(t)}}{P^{(s)}} \right) \tag{45}$$

When $KL(P^{(t)}||P^{(s)}) = 0$, then no information is lost when $P^{(s)}$ is used to approximate $P^{(t)}$. The larger $KL(P^{(t)}||P^{(s)})$, the more information that is lost when $P^{(s)}$ is used to approximate $P^{(t)}$. Therefore, the more unrelated these distributions are. Below we connect the concept of KL-divergence, which we use in practice to determine relatedness of tasks before transfer, to the concept of $\mathcal{F}$-related probability distributions used to derive the bounds in Theorem 4 above. Note that we analyse the information required to transform $P^{(s)}$ into $P^{(t)}$ since in our framework the KL-divergence is determined before transferring forward to the target task, as an indication of relatedness of a source hypothesis and this task.

Lemma 1. *Let $P^{(t)}$ and $P^{(s)}$ be two probability distributions on a domain X, such that both $X^{(t)} \in X$ and $X^{(s)} \in X$. Let $\mathbb{H}$ be a hypothesis space over X. Let $\mathcal{F}$ be a set of transformations $f : \{X^{(t)} \to X^{(s)}, X^{(s)} \to X^{(t)}\}$, such that f is smooth, transitive and acts as a group over $\mathbb{H}$. Let $P^{(t)}$ and $P^{(s)}$ be related if one can be generated from the other by applying some $f \in \mathcal{F}$, such that $P^{(s)} = f[P^{(t)}]$ or $P^{(t)} = f[P^{(s)}]$. Let the KL-divergence when $P^{(s)}$ is used to approximate $P^{(t)}$, $KL(P^{(t)}||P^{(s)})$, be defined as in Eq. 45.*

When $KL(P^{(t)}||P^{(s)}) \to 0$, then $\mathcal{F} \neq \emptyset$ and vice-versa. On the other hand, when $KL(P^{(t)}||P^{(s)}) \to u$, with $u \gg 0$, the probability that $\mathcal{F} = \emptyset \to 1$, and vice-versa.

Proof. For $\mathcal{F} \neq \emptyset$ this implies that there exists an $f \in \mathcal{F}$ such that $P^{(t)} = f[P^{(s)}]$. By Lemma 2 in Ben-David and Borbely (2008), there also exists an $f' \in \mathcal{F}$ such that $P^{(s)} = f'[P^{(t)}]$. $f' \in \mathcal{F}$ will contain the information required to transform $P^{(s)}$ into $P^{(t)}$. From Definition 5, the existence of such $\mathcal{F}$ implies a reduction on the hypotheses space from $\mathbb{H}$ to $\mathcal{H}$, where $\mathcal{H} = \{[\hbar] : \hbar \in \mathbb{H}\}$ is a family of hypothesis spaces which is the family of all equivalence classes of $\mathbb{H}$ under $\sim_{\mathcal{F}}$. The smaller $\mathcal{F}$, the smaller each $[\hbar]$, since $[\hbar] = \{\hbar \circ f : f \in \mathcal{F}\}$. Intuitively, being more restricted the set of functions required to transform

$P^{(s)}$ into $P^{(t)}$, the less divergent $P^{(s)}$ and $P^{(t)}$ are, and therefore the less the amount of information lost when using $P^{(s)}$ directly to approximate $P^{(t)}$, *i.e.* $KL(P^{(t)}||P^{(s)}) \to 0$.

On the other hand, $\mathcal{F} = \emptyset$ implies that there is no $f \in \mathcal{F}$ such that $P^{(t)} = f[P^{(s)}]$ or $P^{(s)} = f'[P^{(t)}]$, and that $\mathcal{F}$ be a group and $\mathbb{H}$ is closed under the action of $\mathcal{F}$, *i.e.* for every $f \in \mathcal{F}$ and every $\hbar \in \mathbb{H}$ then $\hbar \circ f \in \mathbb{H}$. If there is not $\hbar \circ f \in \mathbb{H}$ then there is no information in $\mathbb{H}$ that can be used to transform $P^{(s)}$ into $P^{(t)}$. Therefore, the more divergent $P^{(s)}$ and $P^{(t)}$ are, which implies the larger the amount of information lost when using $P^{(s)}$ to approximate $P^{(t)}$, *i.e.* $KL(P^{(t)}||P^{(s)}) \to u$. □

We have explored the properties of the proposed method for updating a hypothesis $f^{(s)}$ given additional information transferred from a related $f^{(t)}$ learned recently. We now approach our proposed lifelong learning setting where an existing $f^{(s)}$ can be refined several times while more $f^{(t)}$ are learned sequentially.

6.5.3 Sequentiality of Tasks

We first formalise our lifelong learning setting of refining an existing $f^{(s)}$ while target tasks $T^{(t)}$ are observed and learned sequentially.

Definition 7. *Let $P^{(s)}$, $P^{(t)}$, $D^{(s)}$, $D^{(t)}$, $T^{(s)}$ and $T^{(t)}$ as defined in Theorem 4. Let $P^{(t+1)}$ be a target distribution for learning a hypothesis $f^{(t+1)}$ on a task $T^{(t+1)}$ at time $(t+1)$, using a sample $D^{(t+1)}$. Let $\mathbb{H}$ be a hypothesis space over the domain X and $\mathcal{F}$ a set of transformations over that set such that $\mathbb{H}$ is closed under the action of $\mathcal{F}$.*

Given classes $\mathcal{F}$ and $\mathbb{H}$, and labeled samples $D^{(s)}, D^{(t)}, D^{(t+1)}$ for tasks $T^{(s)}, T^{(t)}, T^{(t+1)}$ at time $(t+1)$, our sequential method:

- *Selects $\hbar^*_{(t+1)} \in \mathbb{H}$ that minimises:*

$$\inf_{\hbar_1,\ldots,\hbar_n \in [\hbar]} \left[\left(\inf_{\hbar_1,\ldots,\hbar_n \in [\hbar]} \hat{\mathbb{E}}r^{D^{(s)}}(\hbar) + \Gamma_t \hat{\mathbb{E}}r^{D^{(t)}}(\hbar) \right) + \Gamma_{(t+1)} \hat{\mathbb{E}}r^{D^{(t+1)}}(\hbar) \right] \tag{46}$$

over all $[\hbar] \in \mathbb{H}/\sim_{\mathcal{F}}$, where $\Gamma_t \in [0,1]$ and $\Gamma_{(t+1)} \in [0,1]$ usually small.

- *Selects $\hbar^{\Diamond}{}_{(t+1)} \in [\hbar^*]_{(t+1)}$ that minimises $\mathbb{E}r^{D^{(s)}}(\hbar')$ over all $\hbar' \in [\hbar^*]_{(t+1)}$, and outputs $\hbar^{\Diamond}{}_{(t+1)}$ as the hypothesis for task $T^{(s)}$ at time $(t+1)$.*

By Definition 7, in our special case of a source task $T^{(s)}$, a target task $T^{(t)}$ learned at time t as in Definition 6 and a new target task $T^{(t+1)}$ observed at time $(t+1)$:

Theorem 5. *Let $\{P^{(s)}, P^{(t)}, P^{(t+1)}\}$ be a set of $\mathcal{F}$-related probability distributions, $\{D^{(s)}, D^{(t)}, D^{(t+1)}\}$ random samples representing these distributions, $\mathcal{F}$ a family of domain transformations of the domain X and let $\mathbb{H}$ be a family of binary valued functions on that domain such that $\mathcal{F}$ acts as a group over $\mathbb{H}$. Let $d_{max} = max_{\hbar \in \mathbb{H}} VC\text{-}dim([\hbar]_{\mathcal{F}})$. Let $\hbar^{\Diamond}$ and $[\hbar^*]$ be selected according to Definition 6, and $h^{\#}$ in Theorem 4. Let ϵ_1 be defined as derived in Theorem 4. Then, for every ϵ_3, ϵ_2, $\delta > 0$, and with $\epsilon_3 \leq \epsilon_1$, if:*

$$|D^{(s)}| \geq \frac{64}{\epsilon_3^2}\left[2d_{max}log\frac{12}{\epsilon_3} + log\frac{8}{\delta}\right] \tag{47}$$

and:

$$|D^{(t+1)}| \geq \frac{88}{\epsilon_2^2}\left[2d_{\mathcal{H}}(2)log\frac{22}{\epsilon_2} + \frac{1}{2}log\frac{8}{\delta}\right] \tag{48}$$

then with probability greater than $(1-\delta)$*:*

$$\mathbb{E}r^{P^{(s)}}(\hbar^{\Diamond}{}_{(t+1)}) \leq \inf_{\hbar \in \mathbb{H}} \mathbb{E}r^{P^{(s)}}(\hbar) + 2(\epsilon_2 + \epsilon_3) \tag{49}$$

Proof. Let $\hbar^{\#}_{(t+1)}$ be the best $P^{(s)}$ label predictor at time $(t+1)$ in $\mathbb{H}$, *i.e.* $\hbar^{\#}_{(t+1)} = \arg\min_{\hbar \in \mathbb{H}} \mathbb{E}r^{P^{(s)}}(\hbar)$. Let $[\hbar^{*}_{(t+1)}]$ be the equivalence class picked according to Definition 7. By the choice of $\hbar^{*}_{(t+1)}$:

$$\begin{aligned} &\inf_{\hbar_1,\dots,\hbar_n \in [\hbar^{*}_{(t+1)}]} \left[\left(\inf_{\hbar_1,\dots,\hbar_n \in [\hbar^{*}]} \hat{\mathbb{E}}r^{D^{(s)}}(\hbar) + \Gamma_t \hat{\mathbb{E}}r^{D^{(t)}}(\hbar)\right) + \Gamma_{(t+1)}\hat{\mathbb{E}}r^{D^{(t+1)}}(\hbar)\right] \\ &\leq \inf_{\hbar_1,\dots,\hbar_n \in [\hbar^{\#}_{(t+1)}]} \left[\left(\inf_{\hbar_1,\dots,\hbar_n \in [\hbar^{\#}]} \hat{\mathbb{E}}r^{D^{(s)}}(\hbar) + \Gamma_t \hat{\mathbb{E}}r^{D^{(t)}}(\hbar)\right) + \Gamma_{(t+1)}\hat{\mathbb{E}}r^{D^{(t+1)}}(\hbar)\right] \end{aligned} \tag{50}$$

Based on Theorem 2 in Ben-David and Borbely (2008), by adding a new task $T^{(t+1)}$:

$$\left|\mathbb{E}r^{P^{(s)}}([\hbar_{(t+1)}]) - \inf_{h_1,\dots,h_n \in [\hbar_{(t+1)}]} \left(\hat{\mathbb{E}}r^{D^{(s)}}, \hat{\mathbb{E}}r^{D^{(t)}}, \hat{\mathbb{E}}r^{D^{(t+1)}}\right)\right| \leq \epsilon_3 \tag{51}$$

with:

$$\begin{aligned} &\inf_{h_1,\dots,h_n \in [\hbar_{(t+1)}]} \left(\hat{\mathbb{E}}r^{D^{(s)}}, \hat{\mathbb{E}}r^{D^{(t)}}, \hat{\mathbb{E}}r^{D^{(t+1)}}\right) \\ &= \inf_{h_1,\dots,h_n \in [\hbar_{(t+1)}]} \frac{1}{2}\left[\left(\inf_{h_1,\dots,h_n \in [\hbar]} \frac{1}{2}(\hat{\mathbb{E}}r^{D^{(s)}}(\hbar) + \Gamma_t \hat{\mathbb{E}}r^{D^{(t)}}(\hbar))\right) + \Gamma_{(t+1)}\hat{\mathbb{E}}r^{D^{(t+1)}}(\hbar)\right] \end{aligned} \tag{52}$$

then with probability greater than $(1-\delta/2)$:

$$\begin{aligned} &\inf_{\hbar_1,\dots,\hbar_n \in [\hbar^{\#}_{(t+1)}]} \left[\left(\inf_{\hbar_1,\dots,\hbar_n \in [\hbar^{\#}]} \hat{\mathbb{E}}r^{D^{(s)}}(\hbar) + \Gamma_t \hat{\mathbb{E}}r^{D^{(t)}}(\hbar)\right) + \Gamma_{(t+1)}\hat{\mathbb{E}}r^{D^{(t+1)}}(\hbar)\right] \\ &\leq \mathbb{E}r^{P^{(s)}}([\hbar^{\#}_{(t+1)}]) + \epsilon_3 \end{aligned} \tag{53}$$

and:

$$\begin{aligned} &\mathbb{E}r^{P^{(s)}}([\hbar^{*}_{(t+1)}]) \\ &\leq \inf_{\hbar_1,\dots,\hbar_n \in [\hbar^{*}_{(t+1)}]} \left[\left(\inf_{\hbar_1,\dots,\hbar_n \in [\hbar^{*}]} (\hat{\mathbb{E}}r^{D^{(s)}}(\hbar) + \Gamma_t \hat{\mathbb{E}}r^{D^{(t)}}(\hbar))\right) + \Gamma_{(t+1)}\hat{\mathbb{E}}r^{D^{(t+1)}}(\hbar))\right] + \epsilon_3 \end{aligned} \tag{54}$$

Similar to Theorem 4, combining the inequalities above, with probability greater than $(1-\delta/2)$:

$$\mathbb{E}r^{P^{(s)}}([\hbar^{*}_{(t+1)}]) \leq \mathbb{E}r^{P^{(s)}}([\hbar^{\#}_{(t+1)}]) + 2\epsilon_3 \tag{55}$$

On the other hand, since by Theorem 2 in Ben-David and Borbely (2008), at time t:

$$|D^{(s)}| \geq \frac{88}{\epsilon_1^2}\left[2d_{\mathcal{H}}(2)log\frac{22}{\epsilon_1}+\frac{1}{2}log\frac{4}{\delta}\right] \tag{56}$$

whilst at time $(t+1)$:

$$|D^{(s)}| \geq \frac{88}{\epsilon_3^2}\left[2d_{\mathcal{H}}(3)log\frac{22}{\epsilon_3}+\frac{1}{3}log\frac{4}{\delta}\right] \tag{57}$$

and from Proposition 1 in Ben-David and Borbely (2008):

$$d_{\mathcal{H}}(3) \leq d_{\mathcal{H}}(2) \tag{58}$$

i.e. adding more tasks can only benefit when these tasks are related, and since $|D^{(s)}|$ at time $(t+1)$ is smaller or equal to $|D^{(s)}|$ at time t, then:

$$\epsilon_3 \leq \epsilon_1 \tag{59}$$

Finally, since $\hbar^{\Diamond}_{(t+1)} \in \hbar^{*}_{(t+1)}$, with probability greater than $(1-\delta/2)$, $\hbar^{\Diamond}_{(t+1)}$ will have a $P^{(s)}$ error within $2\epsilon_2$ of the best hypothesis there, *i.e.* $\mathbb{E}r^{P^{(s)}}([\hbar^{*}_{(t+1)}])$. □

6.5.4 Computational Complexity in Practice

Schölkopf et al. (2000) provide additional theoretical guarantees on ν-SVM for classification problems. In terms of computational complexity, solving the problem formulated for **HRSVM** includes the cost of learning functions describing subspaces of shared knowledge and of solving the corresponding optimization problem. A one-class SVM function is $O(n^3)$, with n the number of training examples (*i.e.* pairs of source and target support vectors). For each of these functions, $n = 2$. For m local functions the complexity is $O(2^3 m)$. Refining a source hypothesis has the complexity of learning a ν-SVM, which similar to regular SVM problems can be $i \times O(l)$, with i iterations and l source support vectors, if data is cached (Chang & Lin, 2011).

6.6 Discussion

In this section we introduced **HRSVM**, a method that aims to refine knowledge of previous SVM tasks while learning new SVM tasks. The method pursues the objective of supervised learning systems that become more knowledgeable while learning more tasks. This has been a long-standing concern of the machine learning (Thrun, 1996) and artificial intelligence communities (Robins, 1995). The proposed method relies on transferring backward selected knowledge from a task learned recently to existing SVM hypotheses which are refined using this new knowledge. **HRSVM** differs from existing methods in how the knowledge is transferred, and how this knowledge is used to refine existing hypotheses.

Table 3 summarises previous research that has studied the problem of refining existing knowledge by transferring backward or by similar mechanisms, and compares the proposed **HRSVM** method to existing research. We focus on two aspects: what knowledge is used for transfer or refinement (what to transfer), and how this knowledge is used by the corresponding method (how to transfer).

Table 3: Methods on transferring backward and similar approaches, including **HRSVM**

.

Method	**Approach**	**Knowledge transferred/used (what to transfer)**	**How is knowledge used (how to transfer)**
EBNN (Thrun, 1996)	A meta-layer (meta-network) of knowledge. examples of each function.	Invariances between training with respect to the invariances.	Calculating derivatives of the training examples
ELLA (Ruvolo & Eaton, 2013b)	Extension of a multitask method to the sequential setting.	Shared parameters between learning tasks.	A new task is learned from the existing set of shared parameters and specific parameters for that task.
CL (Fei et al., 2016)	To update a source hypothesis by training it from scratch.	Training examples from the target task.	Target training examples are used as additional negative examples for the existing tasks.
PNN (Rusu et al., 2016)	A single adaptable network.	Weights of the network.	Learning lateral connections of features.
EWC (Kirkpatrick et al., 2017)	A single adaptable network.	Weights of the network.	Slowing down learning of selected weights.
DEN (Yoon et al., 2017a)	A single adaptable network.	Units of the network.	The network is expanded.
Pseudo-Recursal (Atkinson et al., 2018a)	A single adaptable network.	Units of the network.	Previous tasks are re-trained on generated training examples.
OWM (Zeng et al., 2019)	A single adaptable network.	Units of the network.	A new task is learned with weights. orthogonal to existing weights.
HRSVM	Several SVM hypotheses, one for each task.	Selected support vectors that represent shared knowledge.	A source SVM hypothesis is refined using this new knowledge.

7. Experiments and Results of Selective Knowledge Transfer

We test the lifelong learning setting where a sequence of T binary classification tasks from a problem domain are learned. We evaluate if the performance of these systems improves as more tasks are learned. We evaluate **HRSVM**, which according to Algorithm 2 first executes **AccGenSVM** to learn a target task and to collect tuples to be used for refining an existing hypothesis. Software and data used for the experiments is available online (Benavides-Prado, Koh, & Riddle, 2020). We also evaluate ELLA (Ruvolo & Eaton, 2013b), CL (Fei et al., 2016), and using **AccGenSVM** only and learning sequentially with no transfer (BL). For ELLA we use their software available online (Ruvolo & Eaton, 2013a). CL is reimplemented according to their hypothesis refinement algorithm. **HRSVM** is implemented on top of **AccGenSVM**.

To introduce the kinds of results that we aim to explore, we start with experimental results of our method for the abstract example in Figure 6. We generate 200 training examples for each $\{f_1^{(s)},\ldots,f_{11}^{(s)}\}$ problem, 100 positive and 100 negative. Each problem is 10-dimensional, as an extension of the 2-dimensional problem in the example. We randomly add 40% noise to the problem to be refined ($f_1^{(s)}$) and 30% noise to the other problems, by shuffling labels on their corresponding samples. The test set is generated as shown in Figure 6, with 400 examples for each problem, 200 positive and 200 negative. This process is repeated 30 times. For each repetition, the hypothesis $f_1^{(s)}$ will be refined after learning each of the remaining hypotheses sequentially. We use **AccGenSVM** to learn each hypothesis sequentially, starting with $f_1^{(s)}$. The order for learning $\{f_2^{(s)},\ldots,f_{11}^{(s)}\}$ is randomised for each repetition.

For transfer forward using **AccGenSVM** we set the KL-divergence threshold (KL) to 0.4 and the number of nearest-neighbours (nn) to 2. For refining $f_1^{(s)}$ using **HRSVM**, the parameter ν is set to the maximal feasible value (Chen et al., 2005), and $\Gamma = 0.01$. Figure 7 shows that the mean accuracy of $f_1^{(s)}$ can be continuously improved when shared knowledge from each $\{f_2^{(s)},\ldots,f_{11}^{(s)}\}$ is used for refinement, as proposed in our method. This improvement adds to 31.8% after learning all tasks.

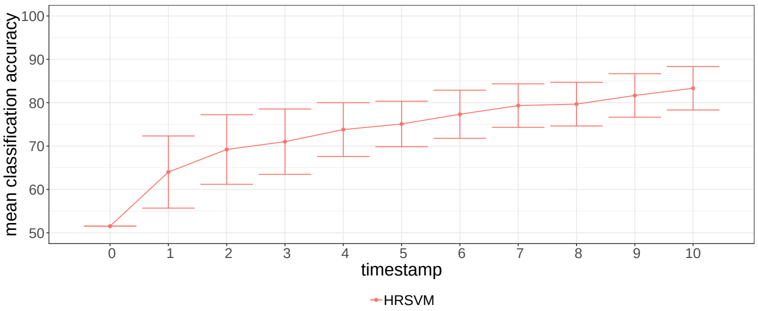


Figure 7: Mean accuracy of hypothesis $f_1^{(s)}$ in the abstract example. Error bars show 95% confidence intervals.

7.1 Datasets and Experimental Set-Up

We generate two synthetic datasets, one of hyperplane problems and one of RBF concepts, composed of 500 tasks each. We also evaluate three real-world datasets. Each dataset is an independent learning system of the form shown in Figure 1. For synthetic hyperplanes we randomly generate 500 problems of 100 features with values in the range $[0, 1]$, using an existing generator (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot, & Duchesnay, 2011). We generate 1,000 random examples for each hyperplane problem. We repeatedly extract training (10%) and test samples (30%) without replacement for each problem, 30 times. We add 40% noise by shuffling training labels. We reimplement an existing method to generate synthetic RBF concepts (Bifet, Holmes, Kirkby, & Pfahringer, 2010). We generate 100 centroids, defined by a random center of 100 features in the range $[0, 1]$, and a standard deviation in the same range. We then generate 1,000 random examples for each RBF concept. We repeatedly extract training (10%) and test samples (30%) without replacement for each class, 30 times, and compose balanced binary classification problems of each RBF concept vs. rest. We add 40% noise by shuffling training labels. We also experiment with 20newsgroups (Mitchell, 1997), CIFAR-100 (Krizhevsky & Hinton, 2009), and a randomly selected ImageNet subset of 500 classes (Deng, Dong, Socher, Li, Li, & Fei-Fei, 2009a). We sample 10% training and 30% test sets, and compose binary classification tasks of each class vs. rest, 30 times.

HRSVM is applied after learning a target task using **AccGenSVM**. For each dataset we train half of the hypotheses as initial sources using a C-SVM (Bottou & Lin, 2007), with $C = 1$, before starting transferring forward with **AccGenSVM**. Synthetic hyperplane tasks are trained with a linear kernel. Synthetic RBF tasks are trained with an RBF kernel and $\gamma = 0.1$ to make them subject to refinement. 20newsgroups, CIFAR and ImageNet tasks are trained with RBF kernels and $\gamma = 1/d$, with d the number of features. We learn one target task at a time. The KL-divergence threshold (KL) of **AccGenSVM** is selected by grid-search on a 5% validation set, with values in $\{0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$. For synthetic hyperplane $KL = 0.3$, for synthetic RBF $KL = 0.45$, for 20newsgroups $KL = 0.5$, for CIFAR-100 and for ImageNet $KL = 0.3$. The number of nearest neighbours is set to 2, similar to Benavides-Prado et al. (2017). **HRSVM** performs hypothesis refinement using the same SVM parameters as the initial sources, ν is set to the maximal feasible value

(Chen et al., 2005) and $\Gamma = 0.01$ in all cases. The sequential learner with no transfer, BL, is trained using C-SVM with $C = 1$.

For ELLA, we tune the number of latent components using grid-search on a 5% validation set, with values in $\{0.05, 0.10, 0.15, 0.20, 0.25\}$, as a percentage of the total number of features. For the sparsity level, we select the optimal value from $\{0.05, 0.1, 0.2, 0.5, 0.8, 1\}$. Best values for the percentage of latent components are: synthetic hyperplane (0.10), synthetic RBF (0.25), 20newsgroups (0.10), CIFAR-100 (0.25), ImageNet (0.20). For all datasets, the sparsity level is set to 1. For datasets of more than 200 features, 200 features are first extracted using PCA[1]. For CL we tune the similarity threshold using grid-search on a 5% validation set, with values in $\{0.10, 0.15, 0.20, 0.25, 0.30\}$. Best values for each dataset are: synthetic hyperplane (0.15), synthetic RBF (0.15), 20newsgroups (0.20), CIFAR-100 (0.20), ImageNet (0.20). The task order is randomized for the 30 repetitions.

We now analyse results of **HRSVM** and counterpart methods in terms of the performance achieved by theses methods while learning tasks sequentially, the convergence rate for finding a solution to the hypothesis refinement problem and the performance for different number of tasks of the proposed method and the counterpart continual learning method DEN (Yoon et al., 2017a).

7.2 Performance Improvement

Figure 8 shows the mean classification accuracy for tasks learned sequentially. The accuracy at each timestamp summarises the accuracy of refined hypotheses, the target hypothesis recently learned and the source hypotheses that remain unchanged. Since ELLA and CL can not be initialised with a number of sources, performances at t_0 are their performances after learning half of the tasks sequentially. **HRSVM** continuously improves performance across timestamps in the synthetic datasets. This denotes that **HRSVM** achieves better performance for systems composed of several related tasks. The gain in performance derived from the proposed hypothesis refinement method can be observed since learning target tasks using only **AccGenSVM** does not encourage increasing performance along the sequence. For real-world data, similar to ELLA, **HRSVM** encourages retention regardless of the number of tasks, with slight improvement. Refinement in these datasets may be more limited due to the presence of more varied and possibly unrelated classes. The improvement in mean classification accuracy after all tasks are learned with **HRSVM** is: synthetic hyperplane 24.4%, synthetic RBF 12.7%, 20 newsgroups 3.7%, CIFAR-100 2.6% and Imagenet 1.2%. A slight improvement may be due to better targets learned by **AccGenSVM**. CL is able to increase the performance at some timestamps, although seems more unstable, possibly due to the change in the number of examples of the negative class and the relatedness of the positive and the negative classes especially in the synthetic datasets. ELLA maintains constant accuracy while learning these tasks, which is in general lower than the counterpart methods. This may be due to the base learner used by ELLA, which is a logistic regression in the ELLA implementation.

Figure 9 shows the number of hypotheses refined by **HRSVM** at each timestamp. During t_1 the largest number of hypotheses are refined, since the full set of initial hypotheses is subject to refinement. For the synthetic datasets, for example, 250 hypotheses are subject

[1] The implementation of ELLA allows up to 200 features.

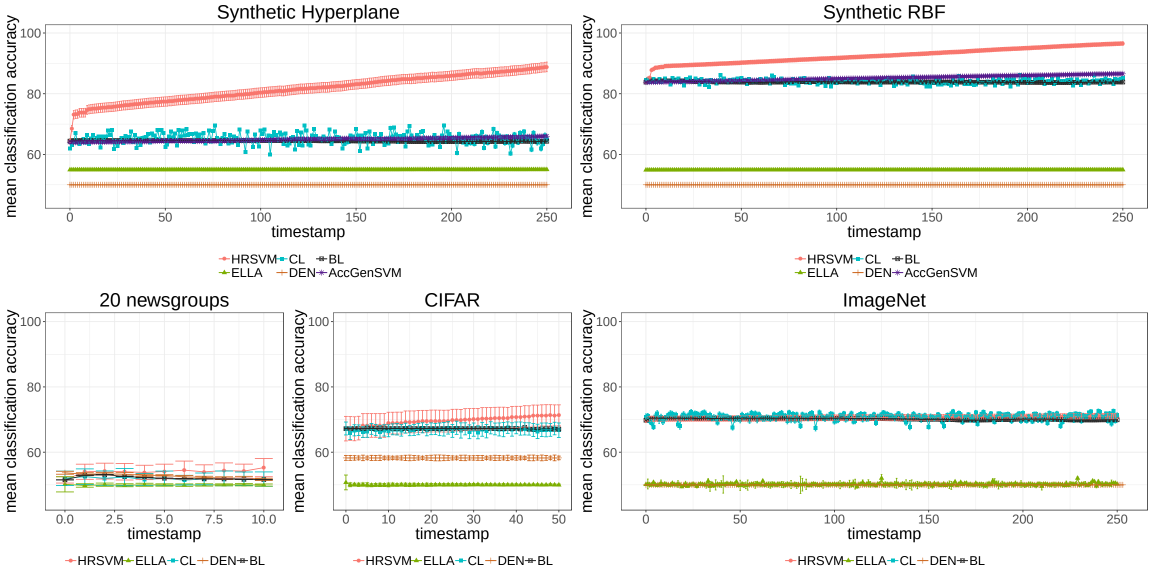


Figure 8: Mean classification accuracy at each timestamp. Error bars show 95% confidence intervals.

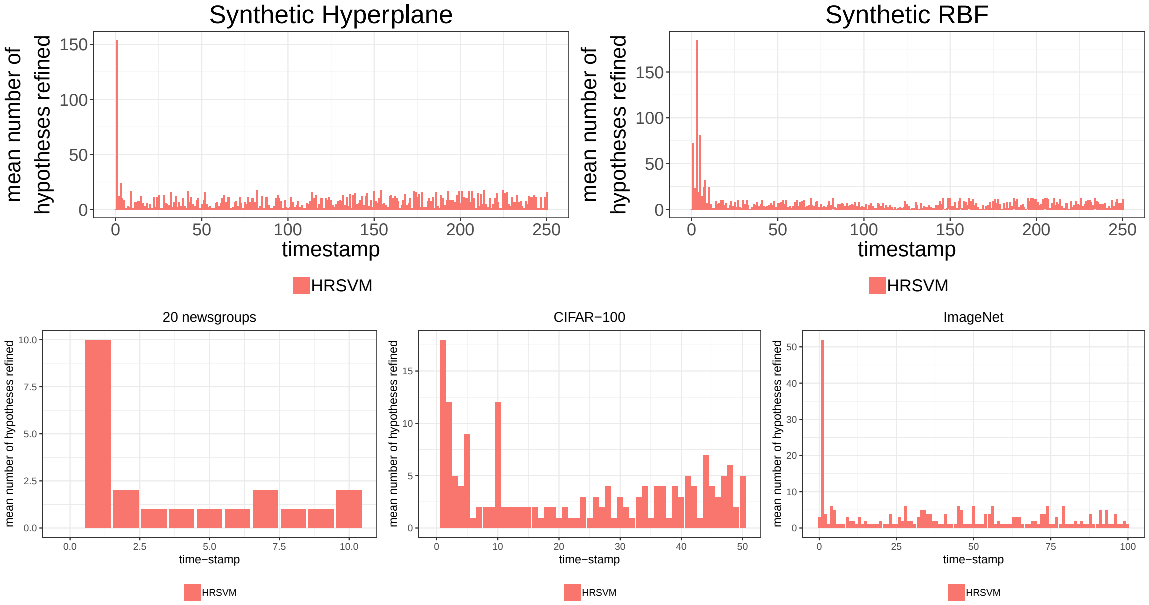


Figure 9: Number of hypotheses refined at each timestamp.

to refinement for the first time. This is related to a large increase in the performance at this timestamp (t_1 in Figure 8). A varying number of hypotheses are refined at subsequent timestamps.

Table 4: Mean number of iterations to converge to a solution, for the hypotheses refined at each timestamp on the synthetic datasets. $--$ denotes non-applicable timestamps for the corresponding dataset.

	Synthetic Hyperplane		**Synthetic RBF**	
t	**CL**	**HRSVM**	**CL**	**HRSVM**
0	289.4 ± 4.9	289.46 ± 4.9	153.1 ± 0.9	153.1 ± 0.9
3	500.0 ± 26.3	28.5 ± 4.7	194.1 ± 3.4	19.1 ± 1.5
6	526.3 ± 23.2	27.1 ± 5.0	194.4 ± 2.9	18.6 ± 2.3
9	517.6 ± 24.8	26.8 ± 5.2	191.9 ± 9.2	17.3 ± 2.2
12	512.1 ± 18.58	24.7 ± 5.6	195.2 ± 2.3	17.9 ± 3.4
15	469.2 ± 21.7	62.0 ± 15.8	193.6 ± 2.1	16.4 ± 2.8
18	494.3 ± 15.5	24.3 ± 6.0	193.8 ± 2.0	16.8 ± 4.0
21	480.2 ± 14.2	53.0 ± 14.1	197.4 ± 1.9	16.1 ± 2.2
25	499.9 ± 20.3	20.5 ± 5.3	194.6 ± 4.3	14.7 ± 1.6

Table 5: Cost measured as the mean number of iterations to converge to a solution, for the group of hypotheses refined at each timestamp on the real-world datasets. $--$ denotes non-applicable timestamps for the corresponding dataset.

	20 newsgroups		**CIFAR-100**		**ImageNet**	
t	**CL**	**HRSVM**	**CL**	**HRSVM**	**CL**	**HRSVM**
0	100.0 ± 0.0	100.0 ± 0.0	190.0 ± 6.9	190.0 ± 6.9	108.6 ± 1.7	108.6 ± 1.7
3	191.7 ± 6.3	13.4 ± 6.6	188.5 ± 5.7	49.4 ± 3.5	192.5 ± 5.1	11.5 ± 2.7
6	193.7 ± 2.2	11.7 ± 3.4	187.0 ± 5.7	48.8 ± 1.2	198.0 ± 9.5	10.7 ± 10.1
9	194.7 ± 0.3	13.9 ± 1.8	186.0 ± 5.7	36.7 ± 3.1	198.7 ± 11.7	10.9 ± 2.2
12	$--$	$--$	187.7 ± 5.5	44.5 ± 4.9	200.0 ± 6.1	10.8 ± 2.2
15	$--$	$--$	187.5 ± 4.9	32.1 ± 4.1	201.9 ± 13.7	10.6 ± 2.0
18	$--$	$--$	185.9 ± 5.5	22.1 ± 5.0	201.1 ± 12.3	10.6 ± 2.2
21	$--$	$--$	189.0 ± 5.0	29.7 ± 4.6	204.1 ± 9.0	10.6 ± 2.2
25	$--$	$--$	188.5 ± 4.2	23.5 ± 5.6	202.3 ± 7.4	10.4 ± 2.5

7.3 Convergence Rate

We measure the number of iterations required to find a solution to Eq. 24 at each timestamp. The convergence rates are summarised by averaging the number of iterations every 25 timestamps. While **HRSVM** requires a fraction of the number of iterations required to train the initial hypotheses (t_0), CL incurs a greater cost since hypotheses are retrained with more training data. **HRSVM** requires up to 21.4% of the number of iterations when hyperplanes are learned from scratch (at $t = 15$), and up to 12.4% for RBF hypotheses (at $t = 3$). Faster convergence rates can also be achieved for real-world datasets. Tables 4 and 5 show example results for the synthetic and real-world datasets under evaluation at different timestamps.

Deep continual learning We experiment with DEN (Yoon et al., 2017a), a recent competitive continual learning method, using the same datasets and samples described

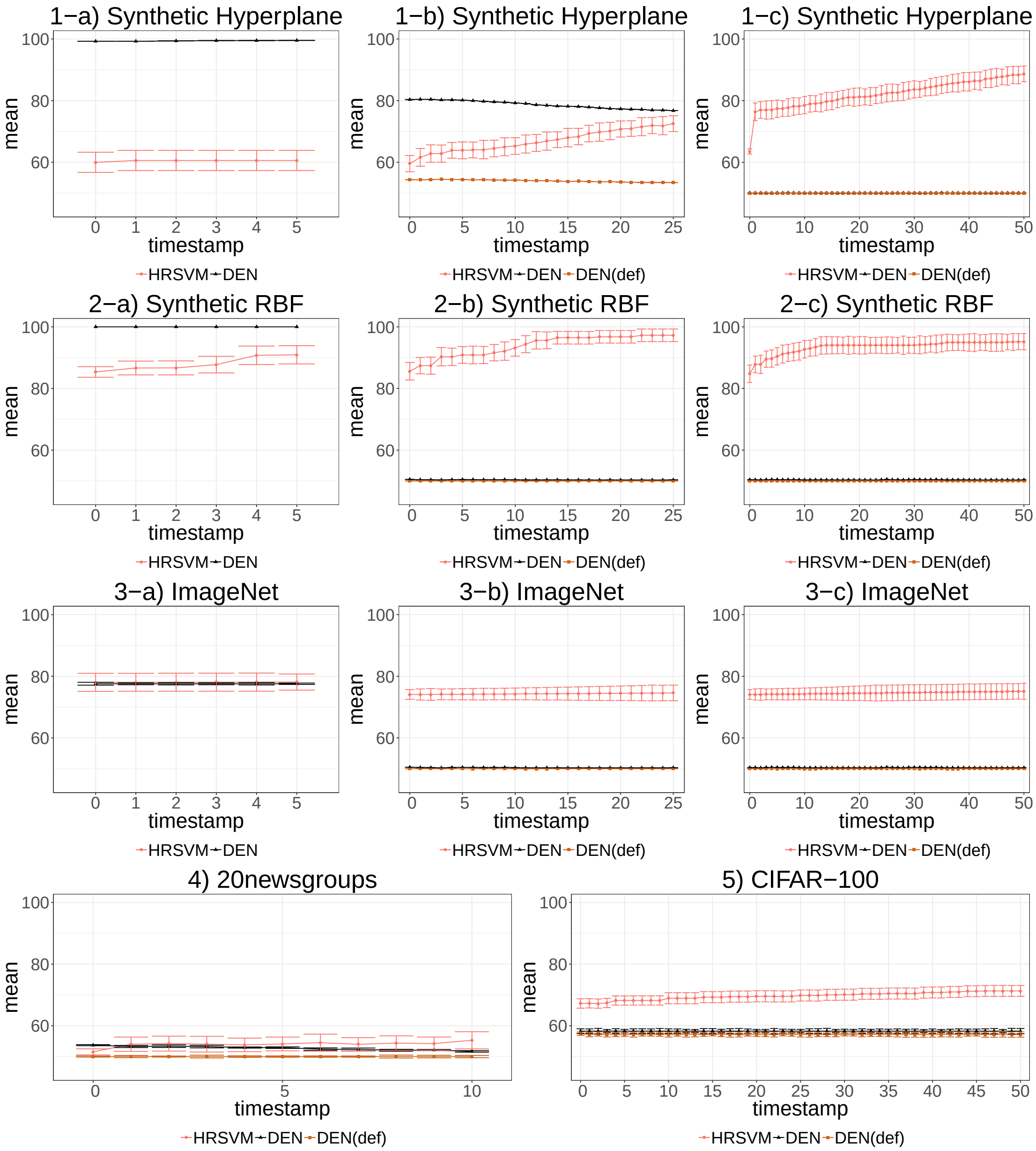


Figure 10: Mean classification accuracy at each timestamp. 1-*a*), 2-*a*), 3-*a*): 10 tasks, 1-*b*), 2-*b*), 3-*b*): 50 tasks, 1-*c*), 2-*c*), 3-*c*): 100 tasks. DEN is a network where the number of units of expansion equals the number of classes. DEN(def) is a network with the default number of units of expansion (10). Error bars show 95% confidence intervals.

previously. The number of hidden layers is set to 2 in all cases[2], with the following number of neurons: for the synthetic datasets 250 and 200 neurons, for ImageNet 500 and 250 neurons, for 20newsgroups 500 and 250 neurons, for CIFAR-100 1,500 and 500 neurons. All networks are FNN that learn binary classification tasks, using the DEN implementation

[2] As allowed by DEN implementation.

(Yoon, Yang, et al., 2017b). After experimentation, we set the values of the parameters: 5,000 maximum iterations, batch size of 500, learning rate of 0.001, L1 sparsity of 0.0001, L2 lambda of 0.0001, group Lasso lambda of 0.001, regularization lambda of 0.5, threshold for dynamic expansion of 0.1, threshold for split and duplication of 0.5. For the number of units of expansion, we experiment with: the default value of 10 and the number of tasks. We test different number of tasks for synthetic hyperplane, synthetic RBF and ImageNet: 10, 50 and 100 tasks. Figure 10 shows results for DEN and **HRSVM**. t_0 is the performance after half of the tasks are learned. Since the number of tasks to learn must be pre-determined in DEN, the performance at t_0 varies depending on the number of tasks to be learned. **HRSVM** does not require the number of tasks as an input. For the synthetic datasets, DEN achieves better performance than **HRSVM** while learning 10 tasks (1-$a, 2$-$a, 3$-a). This performance is constant across timestamps with no indication of refinement. **HRSVM** achieves slightly increasing performance. For 50 tasks (1-$b, 2$-$b, 3$-b), maintaining a single DEN network is more challenging. Yoon et al. (2017) also remarked the challenge of setting an appropriate network capacity when the number of tasks is large. The relation between the number of tasks and the parameters required by the method is yet to be studied. For 100 tasks (1-$c, 2$-$c, 3$-c) DEN faces more challenges to learn these tasks and the performance remains constant. After experimentation we encountered that, the larger the number of tasks, the more difficult for DEN to perform well even on the training examples. **HRSVM** benefits from a larger number of tasks, since there are more chances for knowledge refinement. For ImageNet, 20newsgroups and CIFAR-100 the performance of both **HRSVM** and DEN remains mostly constant over time, when trained using the given parameters.

8. Measurement of Knowledgeable Learning Systems

Chen and Liu (2016) defined the ability to perform continuous learning as one of the core properties of lifelong machine learning systems. Learning continuously should ideally encourage the system to become more knowledgeable. Therefore, a lifelong learning system should demonstrate better performance over time. Measuring this performance is however a challenge since standard performance metrics have been originally designed for learning settings where tasks are executed in isolation.

8.1 Previous Research on Measuring Knowledgeable Lifelong Learners

According to Silver, Yang and Li (2013) and Chen and Liu (2016), a system that learns in the long-term should denote better performance as more tasks are observed and learned. This property was also envisioned for continual learning systems (Ring, 1994). However, due to the infancy of lifelong learning and continual learning, the amount of research that studies how to measure this performance has been very limited. In this section we describe two recent studies that have investigated this problem.

Li and Yang (2015) proposed a lifelong machine learning test to determine if an agent could be categorised as a lifelong learner. A learning agent A would pass this lifelong machine learning test by satisfying two conditions: 1) if its macroaveraging (mean) accuracy is better than a base learner B that learns these tasks separately, at every timestamp t during a sequence of tasks, and 2) if the difference in the macroaveraging accuracy obtained by these two learners becomes larger and larger over time. The test was formally defined as

(Li & Yang, 2015):

$$\forall t : \begin{cases} MA(A^{(t)}) > MA(B^{(t)}) \\ \nabla MA(A^{(t)}) > \nabla MA(B^{(t)}) \end{cases} \tag{60}$$

where $\nabla MA^{(t)} = MA^{(t)} - MA^{(t-1)}$ and $MA^{(t)} = \sum_{t=1}^{T} 1/n^{(t)} \sum_{i=1}^{n^{(t)}} P(y_i^t, f_t(x_i^t))$, with T the number of tasks learned, n the number of examples of all tasks and P a performance metric such as accuracy[3]. Note that multiple lifelong learning agents could be compared indirectly by comparing to the base learner B.

More recently, Diaz-Rodriguez et al. (2018) proposed a set of metrics to measure a variety of characteristics of continual learning systems. These metrics focus on continual learning systems that are based on a single model learned using deep neural networks. Some of these metrics are defined for learning systems composed of any number of tasks. Some other metrics can be measured at any given point during a sequence of tasks. Finally, some of these metrics are to be measured for a particular learning task. The proposed metrics included:

- Accuracy, which considers the mean accuracy of the continual learning system after learning N tasks.
- Backward transfer, which measures the influence that a new task has on the performance of existing tasks. A first attempt at measuring backward transfer was by Lopez-Paz and Ranzato (2017). Diaz-Rodriguez et al. (2018) proposed to measure the average backward transfer after learning N tasks. This metric was further specialized into: 1) remembering and, 2) positive backward transfer. Intuitively, for a continual learning system that does not forget, the former should not decrease dramatically after new tasks are learned, while the latter should increase if the system is able to use new tasks to refine knowledge of tasks learned previously.
- Forward transfer, which measures the influence of N tasks learned previously on the performance of a new task.
- Model size efficiency, which measures the number of parameters added to the network as a result of learning a new task. The authors suggested that an efficient approach should encourage a fewer number of new parameters learned for new tasks. This metric is applicable to continual learners based on neural networks.
- Sample storage size efficiency, which measures the amount of memory needed to store training examples for a particular task. This metric is only applicable for lifelong learners that require to store training examples of the learned tasks.
- Computational efficiency, measured as the number of multiplication and addition operations required to learn a particular task. This metric was designed specifically for continual learners based on neural networks.

[3] Although the original formulation by Li and Yang (2015) referred to a loss function $\mathcal{L}$, the formulation actually was meant to use a performance metric such as P.

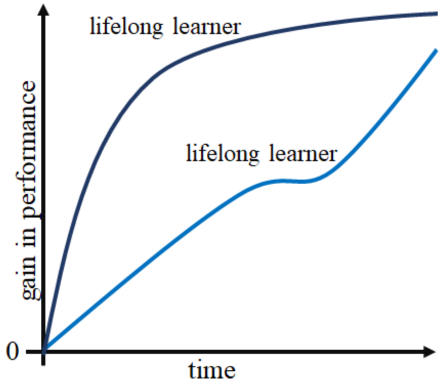


Figure 11: Two lifelong learners with increasing gain in performance.

The proposed metrics are mainly applicable to continual learning systems based on neural networks. In Section 8.2 we propose a more general metric for lifelong learning systems that measure both forward and backward transfer after N tasks. The proposed metric aims to measure how knowledgeable is a learning system after learning these N tasks.

8.2 CGLL

Inspired by the lifelong machine learning test of Li and Yang (2015), we propose **Cumulative Gain of a Lifelong Learner (CGLL)** as an alternative simple metric to determine the cumulative gain in performance achieved by a lifelong learner (Benavides-Prado, Koh, & Riddle, 2018). Intuitively, a lifelong learning system that is becoming more knowledgeable should demonstrate an increasing gain over time. For a sufficiently large number of tasks that are equally relevant for the learning system, such system should denote increasing performance. This demonstrates that the system is sufficiently good at learning new tasks and at refining existing knowledge of previous tasks. This can be potentially achieved if such learning system implements two of the characteristics identified by Chen and Liu (2016): 1) learning new tasks better and 2) performing continuous learning, whilst possibly refining knowledge of existing tasks.

The proposed **CGLL** metric is formulated as:

$$CG(LL)_t = CG(LL)_{(t-1)} + \frac{1}{T_N} \sum_{i=1}^{T_N} P(y_i^{(s)}, f_{it}^{(s)}) - P(y_i^{(s)}, f_{i(t-1)}^{(s)}) \tag{61}$$

with $CG(LL)_0 = 0$. The cumulative gain CG of a lifelong learner LL at a timestamp t depends on the cumulative gain of that learner at the previous timestamp $(t-1)$ and the aggregation of differences in performance for a set of functions or hypotheses $f^{(s)} \in S$ at these timestamps, measured using a performance metric P such as accuracy. Similar to Li and Yang (2015), we propose a test to determine if a lifelong learner can be categorised as a learner that is becoming more knowledgeable. The $CGLL$ test is defined as follows:

$$\forall t : \begin{cases} \text{if} \quad \mathrm{t} > 0 \quad CG(LL)_{(t)} \geq CG(LL)_{(t-1)} \\ \text{if} \quad \mathrm{t} = 0 \quad CG(LL)_{(t)} = 0 \end{cases} \tag{62}$$

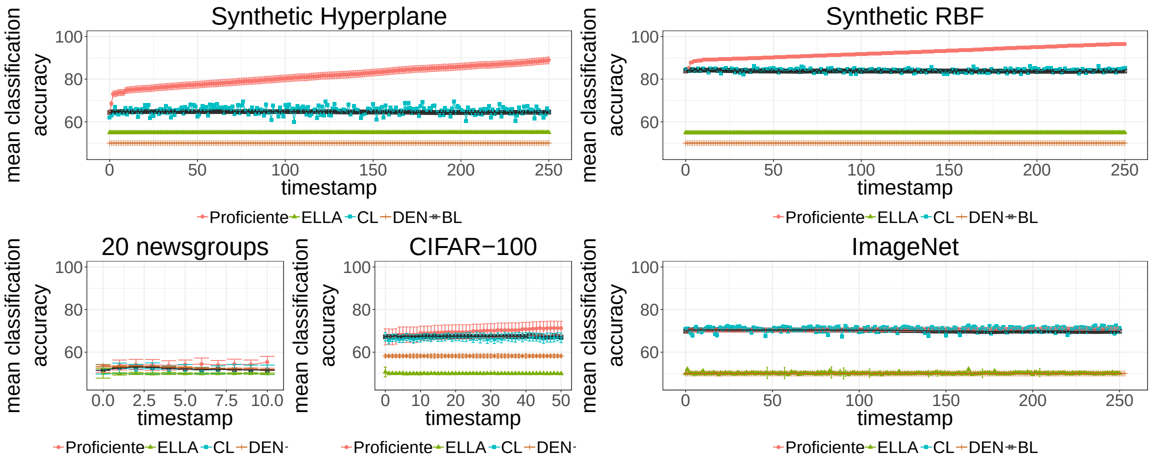


Figure 12: Lifelong learning test (Li & Yang, 2015), evaluated on two synthetic and three real-world datasets. Error bars denote 95% c.i. t_0 denotes performance after half of the tasks have been learned. Note that, while for **Proficiente** t_0 is the initial knowledge base without transfer, for ELLA, CL and DEN t_0 some transfer/refinement may have already occurred since half of the tasks have been learned sequentially using these methods.

i.e. at any time t the cumulative gain of the performance of the system should be at least equal to the previous cumulative gain, *i.e.* $CG(LL)$ is at least monotonically increasing. Note that this restriction could be softened for learning systems that can tolerate a small decrease in performance at some (t) while refining previous knowledge.

Figure 11 shows two examples of lifelong learners that would be categorised as learners that encourage increasing performance over time. Note that the test assumes that all the tasks are equally important. For tasks of similar performance, the results of the test will tend to be steady over time (dark blue line in Figure 11). For learning systems composed of tasks with varied performance, or composed of tasks representing noisy problems, a mechanism such as weighting these tasks differently may be necessary.

8.3 Experimental Evaluation of Knowledgeable Supervised Learning Systems

We evaluate the existing lifelong learning test and the proposed **CGLL** metric on the synthetic and real-world datasets used for experiments in Section 6. To recall, these datasets are: 1) a set of 500 hyperplane binary classification tasks, each of 100 features, 2) a set of 500 binary tasks containing RBF concepts in a one-vs.-rest manner, 3) 20 newsgroups (Mitchell, 1997),4) CIFAR-100 (Krizhevsky & Hinton, 2009) and 5) a subset of 500 ImageNet classes (Deng, Dong, Socher, Li, Li, & Fei-Fei, 2009b). Training and test samples are extracted for these dataset as described in the previous section.

Figure 12 shows the existing lifelong learning test (Li & Yang, 2015) applied using **Proficiente**, ELLA, CL and DEN[4]. BL is an SVM base learner that learns tasks separately.

[4] After extensive experimentation with a variety of parameters we confirmed that for a large number of tasks the performance of DEN remains around 50%. Therefore, for synthetic hyperplane, synthetic RBF and ImageNet we only run up to 100 tasks using this method. We show extended results for proper visualization.

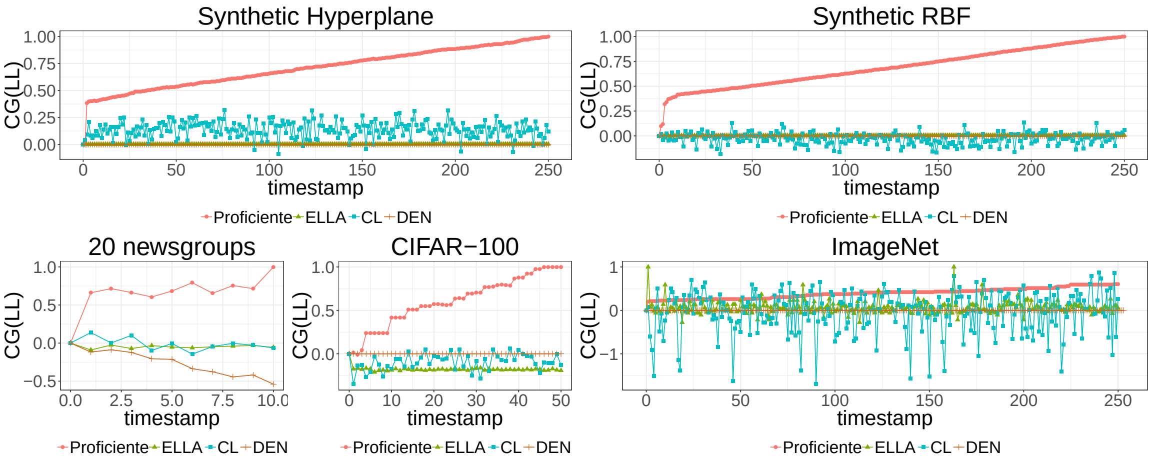


Figure 13: CG(LL) evaluated on two synthetic and three real-world datasets. Error bars denote 95% c.i. t_0 denotes performance after half of the tasks have been learned. Note that, while for **Proficiente** t_0 is the initial knowledge base without transfer, for ELLA, CL and DEN t_0 some transfer/refinement may have already occurred since half of the tasks have been learned sequentially using these methods.

For synthetic hyperplane, this learner is trained with a linear kernel. For synthetic RBF this is trained with an RBF kernel and $\gamma = 0.1$. For 20 newsgroups, CIFAR and ImageNet this learner is trained with an RBF kernel and $\gamma = 1/d$, with d the number of features. In all cases, $C = 1$. We can observe that **Proficiente** achieves increasing performance over time with an increasing gap with respect to the base learner BL, for synthetic hyperplane and synthetic RBF. Methods such as ELLA and DEN find it difficult to achieve increasing performance, whilst their gap with respect to BL remains almost constant over time. CL is volatile over time. For real-world datasets some gap can be achieved also using **Proficiente**, compared to other methods that seem more steady over time.

Figure 13 shows the proposed CGLL metric evaluated for **Proficiente**, ELLA, CL and DEN. For proper visualization, we normalize the cumulative gain over all methods using min-max normalization, with the minimum set to 0. Therefore, in practice the gain of each method is also relative to its counterparts in these experiments. Similar to the previous test, for synthetic hyperplane and synthetic RBF we observe that **Proficiente**, which aims to refine existing knowledge while learning new tasks, can effectively encourage increasing gain. This is an indication of a learning system that is becoming more knowledgeable while learning more tasks. ELLA, CL and DEN find it difficult to pass this test on these datasets. For real-world datasets some gain can be achieved also using **Proficiente**, compared to other methods that seem more unstable when evaluated using the proposed metric.

9. Conclusion and Final Remarks

In this paper we have introduced **Proficiente**, a full framework for learning a sequence of binary classifications tasks observed one at a time. The proposed framework is based on transferring knowledge selectively between tasks learned using SVM. We have demonstrated, both experimentally and theoretically, that transferring selected knowledge forward to new tasks and backward to existing hypotheses representing previous tasks has the potential to encourage learning systems that become more knowledgeable while observing tasks sequentially. Becoming more knowledgeable is a desired ability of machine learning systems in a range of application areas.

An avenue of research that we consider of remarkable importance is the study of methods that pursue similar goals as the method proposed in Section 6. We consider that refining previous knowledge whilst retaining as much of this knowledge as possible can encourage the adoption of long-term learning systems. A particularly interesting avenue of research is to study deep learning methods that aim to exploit knowledge acquired during recent tasks to refine a network in such a way that the set of tasks represented by this network can improve their performance over time. We consider that ideas from previous research in continual learning such as PNN (Rusu et al., 2016) or Pseudo-recursal (Atkinson et al., 2018a) could potentially be adapted to pursue this objective. Nevertheless, there are several fundamental aspects to study in the context of continual learning approaches that aim to become more knowledgeable. One of these aspects is the possibility of maintaining a single network for all the tasks, as opposed to alternatives such as maintaining separate networks for each task similar to the setting proposed in this paper. Another aspect is how to determine which knowledge should be refined, since for deep neural networks models no subset of training examples is maintained as part of the final hypothesis representation, as is the case for SVM. Our current research is focused on exploring these and further challenges in the context of deep neural networks.

More concrete research questions can also be derived from this paper. For example, long-term learning systems that learn several tasks at a time, that need to learn to classify examples in several classes or categories, or that perform selective transfer forward and backward for tasks that use a different feature representation are all feasible and relevant avenues of research. There are additional aspects that can be also studied both theoretically and experimentally. One of these aspects is related to the effects of accumulation of error in lifelong learning systems. As more imperfect tasks are observed and learned, the remaining question is to what extent increasingly knowledgeable lifelong learning systems can be achieved. This effect should be studied especially in systems composed of a large and varied number of tasks.

Finally, we believe it is plausible to study lifelong machine learning systems that aim to become increasingly knowledgeable over time in specific application domains such as dialogue systems, object recognition, and natural language processing. Several problems in these applications domains could substantially benefit by considering learning as a sequential problem where the system aims to become more knowledgeable as more tasks are observed.

Acknowledgments

We would like to thank Prof. Stephen Marsland at Victoria University of Wellington, New Zealand, for his insightful feedback which helped to improve the theoretical analyses presented in this paper. We would also like to thank the reviewers for their insightful comments and valuable feedback which were fundamental for improving this paper.

References

Abu-Mostafa, Y. S. (1995). Hints. *Neural Computation*, *7*(4), 639–671.

Amit, R., & Meir, R. (2018). Meta-learning by Adjusting Priors based on Extended PAC-Bayes Theory. In *ICML*, pp. 205–214.

Atkinson, C., McCane, B., Szymanski, L., & Robins, A. (2018a). Pseudo-Recursal: Solving the Catastrophic Forgetting Problem in Deep Neural Networks. *arXiv preprint arXiv:1802.03875*.

Atkinson, C., McCane, B., Szymanski, L., & Robins, A. (2018b). Pseudo-Rehearsal: Achieving Deep Reinforcement Learning without Catastrophic Forgetting. *arXiv preprint arXiv:1812.02464*.

Aytar, t., & Zisserman, A. (2011). Tabula Rasa: Model Transfer for Object Category Detection. In *ICCV*, pp. 2252–2259. IEEE.

Baxter, J., et al. (2000). A Model of Inductive Bias Learning. *Journal of Artificial Intelligence Research*, *12*(149-198), 3.

Ben-David, S., & Borbely, R. S. (2008). A Notion of Task Relatedness yielding Provable Multiple-Task Learning Guarantees. *Machine Learning*, *73*(3), 273–287.

Benavides-Prado, D., Koh, Y. S., & Riddle, P. (2017). AccGenSVM: Selectively Transferring from Previous Hypotheses. In *IJCAI*, pp. 1440–1446.

Benavides-Prado, D., Koh, Y. S., & Riddle, P. (2018). Measuring Cumulative Gain of Knowledgeable Lifelong Learners. In *NeurIPS Continual Learning Workshop*, pp. 1–8.

Benavides-Prado, D., Koh, Y. S., & Riddle, P. (2019). Selective Hypothesis Transfer for Lifelong Learning. In *IJCNN*, pp. 1–10.

Benavides-Prado, D., Koh, Y. S., & Riddle, P. (2020). HRSVM. https://github.com/nanarosebp/PhDProject/tree/master/HRSVM.

Bengio, Y. (2012). Deep Learning of Representations for Unsupervised and Transfer Learning. In *ICML Workshop on Unsupervised and Transfer Learning*, pp. 17–36.

Beygelzimer, A., Kakadet, S., Langford, J., Arya, S., Mount, D., & Li, S. (2013). FNN: Fast Nearest Neighbor Search Algorithms and Applications. R package version 1.1..

Bifet, A., Holmes, G., Kirkby, R., & Pfahringer, B. (2010). MOA: Massive Online Analysis. *JMLR*, *11*(May), 1601–1604.

Bottou, L., & Lin, C.-J. (2007). Support Vector Machine Solvers. *Large Scale Kernel Machines*, 301–320.

Brazdil, P., Carrier, C. G., Soares, C., & Vilalta, R. (2008). *Metalearning: Applications to Data Mining*. Springer Science & Business Media.

Bremner, A. J., Lewkowicz, D. J., & Spence, C. (2012). *Multisensory Development*. Oxford University Press.

Caruana, R. (1998). Multitask Learning. In *Learning to Learn*, pp. 95–133. Springer.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A Library for Support Vector Machines. *ACM TIST*, *2*(3), 27.

Chen, P.-H., Lin, C.-J., & Schölkopf, B. (2005). A Tutorial on ν-Support Vector Machines. *Applied Stochastic Models in Business and Industry*, *21*(2), 111–136.

Chen, Z., & Liu, B. (2015). Lifelong Machine Learning in the Big Data Era. IJCAI Tutorial.

Chen, Z., & Liu, B. (2016). Lifelong Machine Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *10*(3), 1–145.

Chen, Z., & Liu, B. (2018). Lifelong Machine Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *12*(3), 1–207.

Chen, Z., Ma, N., & Liu, B. (2015). Lifelong Learning for Sentiment Classification. In *ACL*, pp. 750–756.

Cristianini, N., & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*. Cambridge University Press.

Dai, W., Yang, Q., Xue, G.-R., & Yu, Y. (2007). Boosting for Transfer Learning. In *ICML*, pp. 193–200. ACM.

Daumé III, H. (2009). Frustratingly Easy Domain Adaptation. *arXiv preprint arXiv:0907.1815*.

Decoste, D., & Schölkopf, B. (2002). Training Invariant Support Vector Machines. *Machine Learning*, *46*(1), 161–190.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009a). Imagenet: A Large-Scale Hierarchical Image Database. http://image-net.org/download-features.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009b). Imagenet: A Large-Scale Hierarchical Image Database. In *IEEE CVPR*, pp. 248–255. IEEE.

Díaz-Rodríguez, N., Lomonaco, V., Filliat, D., & Maltoni, D. (2018). Don't Forget, There is More than Forgetting: New Metrics for Continual Learning. *arXiv preprint arXiv:1810.13166*.

Duan, L., Tsang, I. W., Xu, D., & Chua, T.-S. (2009). Domain Adaptation from Multiple Sources via Auxiliary Classifiers. In *ICML*, pp. 289–296. ACM.

Fei, G., & Liu, B. (2016). Breaking the Closed World Assumption in Text Classification.. In *HLT-NAACL*, pp. 506–514.

Fei, G., Wang, S., & Liu, B. (2016). Learning Cumulatively to Become More Knowledgeable. In *ACM SIGKDD*, pp. 1565–1574.

Finn, C., Abbeel, P., & Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *arXiv preprint arXiv:1703.03400*.

Ghosn, J., & Bengio, Y. (1997). Multi-Task Learning for Stock Selection. In *NIPS*, pp. 946–952.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. In *ICML*, pp. 513–520.

Gong, B., Shi, Y., Sha, F., & Grauman, K. (2012). Geodesic Flow Kernel for Unsupervised Domain Adaptation. In *IEEE CVPR*, pp. 2066–2073.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Nets. In *NIPS*, pp. 2672–2680.

Hoffman, J., Kulis, B., Darrell, T., & Saenko, K. (2012). Discovering Latent Domains for Multisource Domain Adaptation. In *ECCV*, pp. 702–715. Springer.

Jiang, Y.-G., Wu, Z., Wang, J., Xue, X., & Chang, S.-F. (2018). Exploiting Feature and Class Relationships in Video Categorization with Regularized Deep Neural Networks. *TPAMI*, *40*(2), 352–364.

Jonschkowski, R., Höfer, S., & Brock, O. (2015). Patterns for Learning with Side Information. *arXiv preprint arXiv:1511.06429*.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the National Academy of Sciences*, 201611835.

Krizhevsky, A., & Hinton, G. (2009). Learning Multiple Layers of Features from Tiny Images. *Technical Report*.

Kumar, A., & Daume III, H. (2012). Learning Task Grouping and Overlap in Multi-Task Learning. *arXiv preprint arXiv:1206.6417*.

Kuzborskij, I. (2018). *Theory and Algorithms for Hypothesis Transfer Learning*. Ph.D. thesis, Ecole Polytechnique Fédérale de Lausanne.

Kuzborskij, I., & Orabona, F. (2013). Stability and Hypothesis Transfer Learning. In *ICML*, pp. 942–950.

Kuzborskij, I., & Orabona, F. (2017). Fast Rates by Transferring from Auxiliary Hypotheses. *Machine Learning*, *106*(2), 171–195.

Kuzborskij, I., Orabona, F., & Caputo, B. (2015). Transfer Learning through Greedy Subset Selection. In *ICIAP*, pp. 3–14. Springer.

Lapin, M., Hein, M., & Schiele, B. (2014). Learning Using Privileged Information: SVM+ and Weighted SVM. *Neural Networks*, *53*, 95–108.

Lemke, C., Budka, M., & Gabrys, B. (2015). Metalearning: A Survey of Trends and Technologies. *Artificial Intelligence Review*, *44*(1), 117–130.

Li, L., & Yang, Q. (2015). Lifelong Machine Learning Test. In *AAAI Workshop on "Beyond the Turing Test"*.

Li, X., Zhou, Y., Wu, T., Socher, R., & Xiong, C. (2019). Learn to Grow: A Continual Structure Learning Framework for Overcoming Catastrophic Forgetting. *arXiv preprint arXiv:1904.00310*.

Li, Z., & Hoiem, D. (2017). Learning Without Forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Long, M., Cao, Y., Wang, J., & Jordan, M. I. (2015). Learning Transferable Features with Deep Adaptation Networks. *arXiv preprint arXiv:1502.02791*.

Lopez-Paz, D., & Ranzato, M. (2017). Gradient Episodic Memory for Continual Learning. In *NIPS*, pp. 6467–6476.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In *Psychology of Learning and Motivation*, Vol. 24, pp. 109–165. Elsevier.

Mitchell, T. M. (1997). *Machine Learning*, Vol. 45. Burr Ridge, IL: McGraw Hill.

Mitchell, T. M., & Thrun, S. B. (1993). Explanation-based Neural Network Learning for Robot Control. *NIPS*, 287–287.

Mozafari, A. S., & Jamzad, M. (2016). A SVM-based Model-Transferring Method for Heterogeneous Domain Adaptation. *Pattern Recognition*, *56*, 142–158.

Museum of New Zealand Te Papa Tongarewa (Te Papa), T. O. S. o. N. Z. I., & of Conservation, N. Z. D. (2019 (accessed January 30, 2019)a). The Digital Encyclopaedia of New Zealand Birds - Moorhen. http://nzbirdsonline.org.nz/species/dusky-moorhen.

Museum of New Zealand Te Papa Tongarewa (Te Papa), T. O. S. o. N. Z. I., & of Conservation, N. Z. D. (2019 (accessed January 30, 2019)b). The Digital Encyclopaedia of New Zealand Birds - Pukeko. http://nzbirdsonline.org.nz/species/pukeko.

Niu, L., Li, W., & Xu, D. (2016). Exploiting Privileged Information from Web Data for Action and Event Recognition. *International Journal of Computer Vision*, 1–21.

Oneto, L., Ghio, A., Ridella, S., & Anguita, D. (2015). Shrinkage Learning to Improve SVM with Hints. In *IJCNN*, pp. 1–9. IEEE.

Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. In *CVPR*, pp. 1717–1724.

Pan, S. J., Kwok, J. T., & Yang, Q. (2008). Transfer Learning via Dimensionality Reduction. In *AAAI*, Vol. 8, pp. 677–682.

Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE TKDE*, *22*(10), 1345–1359.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2018). Continual Lifelong Learning with Neural Networks: A Review. *arXiv preprint arXiv:1802.07569*.

Parisotto, E., Ba, J. L., & Salakhutdinov, R. (2015). Actor-Mimic: Deep Multitask and Transfer Reinforcement Learning. *arXiv preprint arXiv:1511.06342*.

Pechyony, D., Izmailov, R., Vashist, A., & Vapnik, V. (2010). SMO-Style Algorithms for Learning using Privileged Information. In *DMIN*, pp. 235–241.

Pechyony, D., & Vapnik, V. (2010). On the Theory of Learning with Privileged Information. In *NIPS*, pp. 1894–1902.

Pechyony, D., & Vapnik, V. (2011). Fast Optimization Algorithms for Solving SVM+. *Statistical Learning and Data Science*, *4*, 3–24.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python - Maximum Margin Separating Hyperplane. http://scikit-learn.org/stable/auto_examples/svm/plot_separating_hyperplane.html.

Pentina, A., & Lampert, C. (2014). A PAC-Bayesian Bound for Lifelong Learning. In *ICML*, pp. 991–999.

Platt, J. (1998). Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines..

Rastegari, M., Farhadi, A., & Forsyth, D. (2012). Attribute Discovery via Predictable Discriminative Binary Codes. In *ECCV*, pp. 876–889. Springer.

Ravi, S., & Larochelle, H. (2017). Optimization as a Model for Few-Shot Learning..

Ring, M. B. (1997). CHILD: A First Step Towards Continual Learning. *Machine Learning*, *28*(1), 77–104.

Ring, M. B. (1994). *Continual Learning in Reinforcement Environments*. Ph.D. thesis, University of Texas at Austin, Texas.

Robins, A. (1995). Catastrophic Forgetting, Rehearsal and Pseudorehearsal. *Connection Science*, *7*(2), 123–146.

Robins, A. (1996). Consolidation in Neural Networks and in the Sleeping Brain. *Connection Science*, *8*(2), 259–276.

Ruder, S. (2017). An Overview of Multi-Task Learning in Deep Neural Networks. *arXiv preprint arXiv:1706.05098*.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., & Hadsell, R. (2016). Progressive Neural Networks. *arXiv preprint arXiv:1606.04671*.

Rusu, A. A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., & Hadsell, R. (2018). Meta-learning with Latent Embedding Optimization. *arXiv preprint arXiv:1807.05960*.

Ruvolo, P., & Eaton, E. (2013a). ELLA. https://github.com/paulruvolo/ELLA.

Ruvolo, P., & Eaton, E. (2013b). ELLA: An Efficient Lifelong Learning Algorithm. *ICML*, *28*, 507–515.

Saenko, K., Kulis, B., Fritz, M., & Darrell, T. (2010). Adapting Visual Category Models to New Domains. In *ECCV*, pp. 213–226. Springer.

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., & Lillicrap, T. (2016). Meta-learning with Memory-Augmented Neural Networks. In *ICML*, pp. 1842–1850.

Schölkopf, B., Burges, C., & Vapnik, V. (1996). Incorporating Invariances in Support Vector Learning Machines. *ICANN*, 47–52.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the Support of a High-Dimensional Distribution. *Neural Computation*, *13*(7), 1443–1471.

Schölkopf, B., Smola, A. J., Williamson, R. C., & Bartlett, P. L. (2000). New Support Vector Algorithms. *Neural Computation*, *12*(5), 1207–1245.

Schölkopf, B., & Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.

Sharmanska, V., Quadrianto, N., & Lampert, C. H. (2013). Learning to Rank using Privileged Information. In *ICCV*, pp. 825–832.

Silver, D. L., Mason, G., & Eljabu, L. (2015). Consolidation using Sweep Task Rehearsal: Overcoming the Stability-Plasticity Problem. In *Canadian Conference on Artificial Intelligence*, pp. 307–322. Springer.

Silver, D. L., & Mercer, R. E. (2000). Selective Transfer of Neural Network Task Knowledge. *The University of Western Ontario (Canada)*.

Silver, D. L., & Poirier, R. (2007). Requirements for Machine Lifelong Learning. In *International Work-Conference on the Interplay between Natural and Artificial Computation*, pp. 313–319. Springer.

Silver, D. L., Yang, Q., & Li, L. (2013). Lifelong Machine Learning Systems: Beyond Learning Algorithms. In *AAAI Spring Symposium: Lifelong Machine Learning*, Vol. 13, pp. 49–53.

Singley, M. K., & Anderson, J. R. (1987). A Keystroke Analysis of Learning and Transfer in Text Editing. *Human-Computer Interaction*, *3*(3), 223–274.

Thrun, S. (1996). *Explanation-Based Neural Network Learning: A Lifelong Learning Approach*. Kluwer Academic Publishers.

Thrun, S., & Pratt, L. (1998). Learning to Learn..

Thrun, S., & Pratt, L. (2012). *Learning to Learn*. Springer Science & Business Media.

Tommasi, T., Orabona, F., & Caputo, B. (2014). Learning Categories from Few Examples with Multi Model Knowledge Transfer. *IEEE TPAMI*, *36*(5), 928–941.

Trevor, H., Robert, T., & JH, F. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction..

Valerio, L., Passarella, A., & Conti, M. (2016). Hypothesis Transfer Learning for Efficient Data Computing in Smart Cities Environments. In *SMARTCOMP*, pp. 1–8. IEEE.

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley, New York.

Vapnik, V., & Izmailov, R. (2015). Learning with Intelligent Teacher: Similarity Control and Knowledge Transfer. In *International Symposium on Statistical Learning and Data Sciences*, pp. 3–32. Springer.

Vapnik, V., & Izmailov, R. (2016). Learning with Intelligent Teacher. In *Symposium on Conformal and Probabilistic Prediction with Applications*, pp. 3–19. Springer.

Vapnik, V., & Vashist, A. (2009). A New Learning Paradigm: Learning using Privileged Information. *Neural Networks*, *22*(5), 544–557.

Wang, Y.-X., & Hebert, M. (2016). Learning by Transferring from Unsupervised Universal Sources. In *AAAI*, pp. 2187–2193.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A Survey of Transfer Learning. *Journal of Big Data*, *3*(1), 1–40.

Yan, Y., Nie, F., Li, W., Gao, C., Yang, Y., & Xu, D. (2016). Image Classification by Cross-Media Active Learning with Privileged Information. *IEEE Transactions on Multimedia*, *18*(12).

Yang, J., Yan, R., & Hauptmann, A. G. (2007). Cross-Domain Video Concept Detection using Adaptive SVMs. In *ACM International Conference on Multimedia*, pp. 188–197. ACM.

Yoon, J., Yang, E., et al. (2017a). Lifelong Learning with Dynamically Expandable Networks. *arXiv:1708.01547*.

Yoon, J., Yang, E., et al. (2017b). Lifelong Learning with Dynamically Expandable Networks. https://github.com/jaehong-yoon93/DEN.

Yoon, J., Kim, T., Dia, O., Kim, S., Bengio, Y., & Ahn, S. (2018). Bayesian Model-Agnostic Meta-learning. In *NIPS*, pp. 7342–7352.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How Transferable are Features in Deep Neural Networks?. In *NIPS*, pp. 3320–3328.

Zeng, G., Chen, Y., Cui, B., & Yu, S. (2019). Continuous Learning of Context-dependent Processing in Neural Networks. *Nature Machine Intelligence*, 364–372.

Zhou, J. T., Xu, X., Pan, S. J., Tsang, I. W., Qin, Z., & Goh, R. S. M. (2016). Transfer Hashing with Privileged Information. *arXiv preprint arXiv:1605.04034*.

Zhuang, F., Cheng, X., Luo, P., Pan, S. J., & He, Q. (2015). Supervised Representation Learning: Transfer Learning with Deep Autoencoders. In *IJCAI*, pp. 4119–4125.